



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

ALEX LALLA
KANTAVIEN RAKENTEIDEN PARAMETRINEN SUUNNITTELU JA
MALLINTAMINEN

Diplomityö

Tarkastaja: professori Sami Pajunen
Tarkastaja ja aihe hyväksytty koulu-
tusvaradekaanin päätöksellä 27.
marraskuuta 2017.

TIIVISTELMÄ

ALEX LALLA: Kantavien rakenteiden parametrinen suunnittelu ja mallintaminen
Tampereen teknillinen yliopisto
Diplomityö, 98 sivua, 6 liitesivua
Joulukuu 2017
Rakennustekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Rakennesuunnittelu
Tarkastaja: professori Sami Pajunen

Avainsanat: parametrinen suunnittelu, parametrinen mallintaminen, algoritmiaivusteinen suunnittelu

Rakenteiden parametrisella suunnittelulla tarkoitetaan menetelmää, jossa rakenteen ominaisuuksiin vaikuttaville parametreille määrätään arvot käsin tai ennalta määriteltyjen sääntöjen avulla siten, että lopputuloksen tuotetaan ominaisuuksiltaan suunniteltu rakenne. Määriteltyt säännöt voivat perustua soveltuvilta osin esimerkiksi suunnitteluohjeisiin tai kokemuseräiseen tietoon. Parametrien arvoja muuttamalla voidaan näin muodostetussa systeemissä muuttaa suunniteltavan rakenteen ominaisuuksia.

Tutkimuksen tavoitteena oli selvittää, miten parametrinen suunnittelu menetelmänä soveltuu rakennesuunnitteluun. Olennaisena osana soveltuvuutta tutkittiin parametrisella suunnittelumenetelmällä saavutettavia etuja ja vastaavasti mahdollisia huonoja puolia verrattuna perinteiseen suunnittelumenetelmään, erityisesti rakennesuunnittelijan näkökulmasta. Tutkimuksessa pyrittiin myös selvittämään, onko parametrinen suunnittelumenetelmän soveltuvuudessa eroja rakenneosakohtaisesti ja miten erilaisia rakennneosia on järkevää suunnitella parametrisella menetelmällä. Lisäksi selvitettiin, onko rakenteiden optimointia tarkoituksenmukaista integroida parametriseen suunnitteluprosessiin.

Tutkimus toteutettiin kahdella menetelmällä. Kirjallisuusselvityksellä selvitettiin parametrinen suunnittelumenetelmän ominaisuuksia ja periaatteita sekä hyödyntämistä eri suunnittelualoilla. Kirjallisuusselvityksessä perehdyttiin myös optimointiin ja kantavien rakenteiden optimointitapoihin. Tietoa parametrinen suunnittelumenetelmän soveltuvuudesta rakennesuunnitteluun kerättiin kvalitatiivisesti pyrkien soveltamaan kirjallisuusselvityksen tuloksia erilaisten rakenneosien ja rakenteiden suunnittelussa.

Tutkimuksessa havaittiin, että soveltamalla parametrista suunnitteluprosessia voidaan rakennesuunnitteluun kuluva aikaa lyhentää, kun tarkastellaan ajanjaksoa luonnossuunnittelusta toteutussuunnitteluun. Parametrisella prosessilla voidaan myös tutkia perinteistä suunnittelumenetelmää suurempi määrä vaihtoehtoja rakenneratkaisuille sekä hyödyntää optimointia tehokkaasti rakenteiden kustannusten minimoinnissa. Ongelmaksi parametrinen suunnittelun soveltamisessa havaittiin, että parametrinen prosessi voi toisaalta johtaa myös pidempään suunnitteluun kuluvaan aikaan. Havaittiin, että suunnitteluun kuluvan ajan määrittäminen etukäteen on haastava tehtävä, jonka tueksi tarvitaan kokemuseräistä tietoa.

Tärkeimpinä tuloksina tutkimuksessa esitetään komponenttipohjainen menetelmä kantavien rakenteiden parametriseen suunnitteluun. Menetelmällä saavutetaan etua erityisesti moniparametrinen rakenteiden, kuten ristikoiden suunnittelussa. Lisäksi tuloksena esitetään, että parametrinen malli on aina suunniteltava, sekä perustelut tälle väitteelle.

ABSTRACT

ALEX LALLA: Parametric Design and Modelling of Structural Frames

Tampere University of Technology

Master of Science Thesis, 98 pages, 6 Appendix pages

December 2017

Master's Degree Programme in Civil Engineering

Major: Structural Engineering

Examiner: Professor Sami Pajunen

Keywords: parametric design, parametric modelling, algorithm aided design

The method where the properties of a structure are modified by a series of parameters whose values are defined either manually or by a set of predefined rules is called parametric design of structures. The predefined rules can be based on for example design guidelines or experiential knowledge. By modifying the parameters in an algorithmic system constructed with these rules and manually modified parameters, the design and its properties can be modified.

The goal in this study was to determine how parametric design as a design method is applicable to structural design process. To achieve this, it was substantive to study the pros and cons of the method compared to the traditional method of structural design, especially from the viewpoint of a structural designer. It was also studied which kind of differences there is in applying parametric design process to different kind of structures and how will the found differences affect the applicability of the process in practical design. In addition, it was studied should optimum design methods be integrated in the parametric design process.

The study was implemented with two methods. A literary review was conducted to find out characteristics and principles of the parametric design method. The usage of parametric methods in other field of design was also studied. With this method, optimum design and the application of optimum design methods to structural design was studied. The information about the suitability of the parametric design method when applying to the design of frame structures was collected with a qualitative method by implementing the results of the literary review.

In this study, it was found that time required for the structural design can be reduced with the use of a parametric design process when considering the span from conceptual design to production design. The parametric design method also allows to explore a wider variety of design alternatives than the traditional design process in the same amount of time. The parametric design approach also makes it simpler and more efficient to apply optimum design methods for reducing the material costs of a project. Some difficulties in applying parametric design process were also observed. The application of parametric design process can lead to extended length of the design process since it is difficult to estimate the required time for each design method without experiential knowledge.

The most important result of the study is the introduction of a component based method for the parametric design process. With this method, benefits can be achieved especially in the design of multiparameter structures such as trusses. In addition, it is shown and justified that parametric models should always be planned before implementation.

ALKUSANAT

Tämä diplomityö on toteutettu A-Insinöörit Suunnittelu Oy:lle. Tarkoituksena on ollut luoda pohja parametristen ja algoritmisten suunnittelumenetelmien hyödyntämiselle todellisissa suunnittelutehtävissä. Vaikka tässä työssä otettiin vasta soveltamisen ensiaskeleita rakennesuunnittelun saralla, menetelmä on osoittanut potentiaalin suunnitteluprosessin tehokkuuden kehittämisessä.

Työn tarkastajaa TTY:lta, Sami Pajusta, haluan kiittää laajoista näkökulmista, jotka auttoivat minua kysymään oikeita kysymyksiä sekä sovittamaan työn sisällön laajempaan kontekstiin. Erityinen kiitos ohjaajalleni Ilari Pirhoselle A-Insinöörit Suunnittelu Oy:stä asiantuntevasta ja innostuneesta ohjauksesta, jolla on ollut merkittävä vaikutus niin työn etenemiseen kuin sen lopulliseen muotoon. Kiitokset myös A-Insinöörit Suunnittelu Oy:n Ville Laineelle mielenkiintoisesta aiheesta ja Villen lisäksi Seppo Nissilälle työn toteuttamisen mahdollistamisesta. Lisäksi haluan kiittää koko Turun toimiston porukkaa, joilla on aina riittänyt aikaa keskustella aiheeseen liittyvistä asioista.

Suurin kiitos kuuluu kuitenkin vaimolleni Katriinalle koko opiskeluajan jatkuneesta tuesta sekä siitä, ettei aika ole tullut pitkäksi vapaa-ajallakaan.

Turussa, 20.12.2017

Alex Lalla

SISÄLLYSLUETTELO

1.	JOHDANTO	1
1.1	Tutkimuksen tausta	1
1.2	Tutkimuksen tavoitteet	1
1.3	Tutkimuksen rajaus	2
2.	TEORIA	4
2.1	Mitä on parametrinen suunnittelu	4
2.2	Parametrit ja reunaehdot	6
2.3	Geometrisen reunaehtotehtävän ratkaisumenetelmät	10
2.4	Algoritmit	12
2.5	Parametrin mallin parametrimäärä	15
3.	OPTIMOINTI	17
3.1	Mitä optimointi tarkoittaa	17
3.2	Optimointi prosessina	17
3.3	Optimointitehtävän formulointi	19
3.4	Optimointi rakennesuunnittelussa	24
3.5	Grasshopper – Galapagos	25
4.	PARAMETRINEN SUUNNITTELU KÄYTÄNNÖN TYÖKALUNA	28
4.1	Muut suunnittelualat	28
4.2	Parametrinen suunnittelu käytännössä	30
4.3	Parametrin suunnittelun taloudellisuus ja tehokkuus	34
4.3.1	Taloudellisuus	34
4.3.2	Tehokkuus	37
4.4	Rakenteen mallintaminen	43
5.	PARAMETRISOINNIN HYÖDYNTÄMINEN RAKENNESUUNNITTELUSSA	
	46	
5.1	Parametrinen mallintaminen	46
5.2	Parametrin mallin suunnittelu	46
5.3	Rakennesien mallintaminen parametrisesti	47
5.3.1	Rakennesien parametrisointi	47
5.3.2	Palkki	49
5.3.3	Pilari	52
5.3.4	Ristikko	54
5.3.5	Liitokset	59
5.3.6	Muut rakennesiat	64
5.4	Parametrinen malli	65
5.5	Yhteistyö muiden suunnittelualojen kanssa	69
5.5.1	Käytännön yhteistyö	69
5.5.2	CASE: Elizabeth Quay Pedestrian and Cyclist Bridge	75
5.6	Optimointi	78
5.6.1	Tutkimusasetelma	78

5.6.2	Geneettinen algoritmi.....	80
5.6.3	Simuloitu jäähdytys.....	84
5.6.4	Yhteenveto	86
5.7	Taloudellisuus	88
6.	YHTEENVETO	91
6.1	Parametrisen suunnittelun hyödyt ja haasteet	91
6.2	Parametrisen mallin suunnittelu, toteutus ja optimointi.....	93
6.3	Tutkimustarpeet tulevaisuudessa	94
	LÄHTEET	96

LIITE A: OPTIMOINTIAJOJEN TULOKSET

KUVALUETTELO

<i>Kuva 1.1. Tutkimuksen tavoitteet.</i>	2
<i>Kuva 2.1. Esimerkki parametrisesta systeemistä.</i>	5
<i>Kuva 2.2. Jatkoa kuvalle 2.1.</i>	6
<i>Kuva 2.3. Elementtijoukon ja ominaisuusjoukon välinen yhteys, kun molemmissa joukoissa on kolme alkiota.</i>	9
<i>Kuva 2.4. Esimerkki visuaalisesta algoritmista.</i>	14
<i>Kuva 3.1. Perinteinen suunnitteluprosessi ja optimointisuunnitteluprosessi (muokattu lähteestä Arora 2004).</i>	19
<i>Kuva 3.2. Galapagos-ratkaisijan valinta-algoritmit (muokattu lähteestä (Rutten 2010)).</i>	26
<i>Kuva 4.1. The Morpheus Hotel (Zaha Hadid Architects)</i>	29
<i>Kuva 4.2. Perinteisen suunnittelumenetelmän ja parametrisen suunnittelun välinen ero.</i>	31
<i>Kuva 4.3. Esimerkki visuaalisesta ohjelmoinnista Grasshopper-ympäristössä.</i>	33
<i>Kuva 4.4. MacLeamyn kuvaaja (CURT 2004).</i>	35
<i>Kuva 4.5. Davisin (2013) parametriselle suunnitteluprosessille modifioima MacLeamyn käyrä.</i>	36
<i>Kuva 4.6. Keskeiset suorituskyvyn mittarit eri tavoin suunnitelmiaan uudelleenhyödyntävissä organisaatioissa (Aberdeen 2007).</i>	38
<i>Kuva 4.7. Esimerkki komponentista Grasshopper-ympäristössä.</i>	40
<i>Kuva 4.8. Esimerkki parametripuun osasta.</i>	43
<i>Kuva 5.1. Parametrinen palkkikomponentti.</i>	49
<i>Kuva 5.2. Palkkikomponentti sahatavaran mitoituksessa.</i>	50
<i>Kuva 5.3. FEM-pohjainen palkkikomponentti.</i>	51
<i>Kuva 5.4. FEM-komponentilla analysoitu ulokepalkki.</i>	51
<i>Kuva 5.5. Pilarikomponentin toiminta.</i>	53
<i>Kuva 5.6. Ristikon parametrinen suunnitteluprosessi.</i>	55
<i>Kuva 5.7. Parametrien lukitseminen säännöillä.</i>	57
<i>Kuva 5.8. Ristikkokomponentti.</i>	58
<i>Kuva 5.9. Ristikkokomponentilla luotuja ristikkoja.</i>	59
<i>Kuva 5.10. Esimerkki liitoksen parametrisesta taulukkosuunnittelusta.</i>	61
<i>Kuva 5.11. Esimerkki laskentamenetelmään perustuvasta liitossuunnittelusta.</i>	62
<i>Kuva 5.12. Esimerkki täysin parametrisesta liitossuunnittelukomponentista.</i>	63
<i>Kuva 5.13. Esimerkki parametrisesta mallista Grasshopper-ympäristössä.</i>	66
<i>Kuva 5.14. Hallialgoritmin toimintaperiaate.</i>	67
<i>Kuva 5.15. Parametrisella mallilla tuotettuja hallin tietomalleja.</i>	68
<i>Kuva 5.16. Yhteistyöprosessi arkkitehdin ja rakennesuunnittelijan välillä samassa parametrisessä mallissa.</i>	70
<i>Kuva 5.17. Yhteistyöprosessi arkkitehdin ja rakennesuunnittelijan välillä, kun kummatkin osapuolet toimivat omissa malleissaan.</i>	71

<i>Kuva 5.18. Annetun geometrian perusteella määritetty ristikkorakenne.....</i>	<i>72</i>
<i>Kuva 5.19. Ristikkoalgoritmin kehitys komponentiksi.</i>	<i>74</i>
<i>Kuva 5.20. Parametrimäärän riippuvuus toisesta parametrasta.</i>	<i>75</i>
<i>Kuva 5.21. Elizabeth Quay Bridge (Archdaily)</i>	<i>76</i>
<i>Kuva 5.22. Parametrinen prosessi case-kohteessa (muokattu lähteestä (Avern-Taplin et al. 2016)).....</i>	<i>77</i>
<i>Kuva 5.23. Optimointiajojen tutkimusasetelma.</i>	<i>79</i>
<i>Kuva 5.24. Optimoitavan ristikon lähtötilanne.....</i>	<i>80</i>
<i>Kuva 5.25. Geneettisellä algoritmilla optimoitujen ristikoiden massojen keskiarvo pysyvien iteraatiokierrosten funktiona.</i>	<i>81</i>
<i>Kuva 5.26. Iteraatiokierrosten lukumäärä pysyvien iteraation funktiona (geneettinen algoritmi).</i>	<i>81</i>
<i>Kuva 5.27. Kaikkien geneettisen algoritmin optimointiajojen iteraatiokierrosten lukumäärä massan funktiona.</i>	<i>82</i>
<i>Kuva 5.28. Skemaattinen kuvio optimointitehtävän luonteesta.</i>	<i>83</i>
<i>Kuva 5.29. Optimoitu massan keskiarvo simuloidulla jäähdytyksellä.....</i>	<i>85</i>
<i>Kuva 5.30. Massojen minimi simuloidulla jäähdytyksellä.....</i>	<i>85</i>
<i>Kuva 5.31. Optimointiajokierrosten tuloksena saadut massat kierrosten funktiona.</i>	<i>86</i>
<i>Kuva 5.32. Optimoitujen ristikkojen massojen otoskeskihajonta.....</i>	<i>87</i>
<i>Kuva 5.33. Optimoitu ristikko sekä vertailuristikko (Mela 2013).....</i>	<i>88</i>

LYHENTEET JA MERKINNÄT

AAD	<i>Algorithms-Aided Design</i> , algoritmiavusteinen suunnittelu. Yleisnimi suunnittelumenetelmille, jotka hyödyntävät (yleensä suunnittelijan itsensä luomia) algoritmeja.
CAD	<i>Computer-Aided Design</i> , tietokoneavusteinen suunnittelu. Yleisnimi tietokoneella käytettäville suunnitteluohjelmille. Käytännössä kaikki modernit suunnitteluohjelmistot ovat CAD-ohjelmistoja.
BES	Betonielementtistandardi, 1970-luvulla Suomessa käyttöönotettu järjestelmä, jolla standardoitiin betonielementtien koot ja liitokset tehokkuuden ja kilpailukyvyn parantamiseksi.
FEM	<i>Finite element method</i> , elementtimenetelmä. Numeerinen ratkaisumenetelmä, jota voidaan soveltaa lujuuslaskelmissa.
GA	<i>Genetic algorithm</i> , geneettinen algoritmi. Ryhmä globaalin haun optimointimenetelmiä, jotka jäljittelevät luonnon evoluutioprosessia. Heuristinen, diskreettiin ja sekalukuoptimointiin soveltuva menetelmä.
PSO	<i>Particle swarm algorithm</i> , parveilualgoritmi. Diskreettiin ja sekalukuoptimointiin soveltuva globaalin haun optimointimenetelmä.
SA	<i>Simulated annealing</i> , simuloitu jäähdytys. Diskreettiin ja sekalukuoptimointiin soveltuva optimointimenetelmä, joka perustuu muutosmahdollisuuksien pienentymiseen optimointiprosessin edetessä.
TS	<i>Tabu search</i> , Tabu-etsintä. Diskreettiin ja sekalukuoptimointiin soveltuva paikallisen haun optimointimenetelmä.

1. JOHDANTO

1.1 Tutkimuksen tausta

Parametrisella suunnittelulla tarkoitetaan suunnittelumenetelmää, jossa rakennus tai rakenneosa ositellaan sellaisiin parametreihin, jotka ovat kriittisiä suunniteltavan kohteen toiminnallisuuden kannalta ja suunnittelutyö tehdään näitä parametreja varioimalla. Parametreja varioimalla, esimerkiksi tietokoneohjelmiston avulla, voidaan tehokkaasti etsiä käyttötarkoitukseen ja tilanteeseen soveltuvia suunnitteluratkaisuja.

Kirjallisuudessa parametrissa suunnittelua käsitellään pääasiassa arkkitehtuurin ja arkkitehtisuunnittelun näkökulmasta. Parametrinen suunnittelumenetelmä on tehokas työkalu erilaisten pintojen ja muotojen muodostamiseen ja tarkasteluun, mikä on erityisesti arkkitehtiteja kiinnostava ominaisuus. Menetelmä mahdollistaa monimutkaisten kolmiulotteisten muotojen entistä helpomman suunnittelun ja näin ollen tällaisen geometrian entistä laajemman hyödyntämisen. Tämä johtaa luonnollisesti siihen, että monimutkaisia muotoja hyödynnetään laajemmassa määrin toteutuvissa rakennuksissa. Tämä aiheuttaa painetta rakennesuunnittelutoimialalle, jossa on perinteisesti pyritty suunnittelemaan mahdollisimman yksinkertaisia, mielellään joko vaaka- tai pystysuoria kantavia rakenteita. Mahdolliset monimutkaisemmat muodot on pyritty toteuttamaan erilaisilla sekundäärirakenteilla. Kun arkkitehtuuri kehittyy kompleksisemmän geometrian suuntaan, myös vaatimukset kantavien rakenteiden geometrialle kasvavat. Nykyaikaiset FEM-ohjelmistot mahdollistavat analyttisiä ratkaisuja monimutkaisempien geometrioiden analysoinnin ja lujuuslaskennan. Monissa tapauksissa puuttuvaksi osuudeksi jää juuri geometrian mallintaminen rakennesuunnittelijan tarpeisiin. Tähän paras ratkaisu lienee vastaava menetelmä, jolla geometria on alun perin luotu.

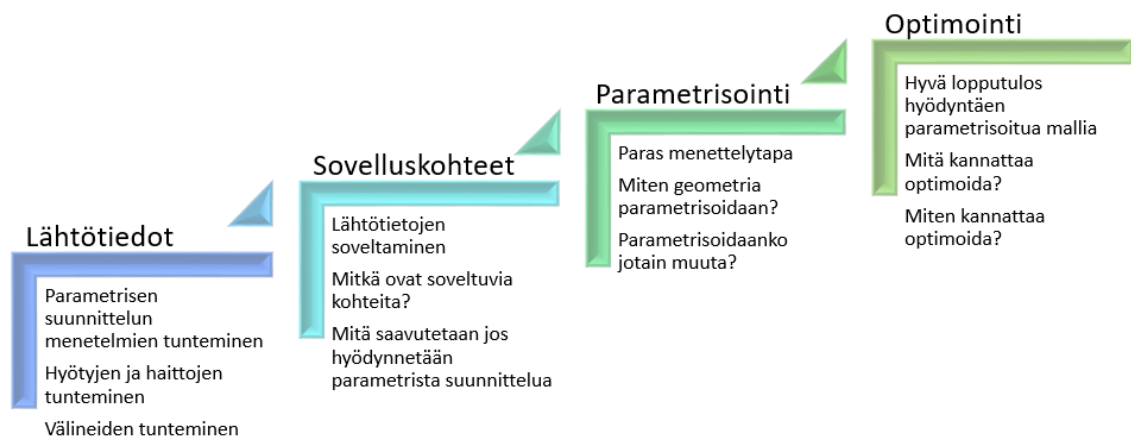
Kun parametrissa suunnittelua tehdään ohjelmistoavusteisesti, voidaan erilaisia suunnitteluratkaisuja tutkia tehokkaasti. Suunniteltaessa komponenttipohjaisesti, suunnitteluprosessia voidaan merkittävästi nopeuttaa yhdistämällä luonnossuunnittelu, lujuuslaskenta sekä toteutussuunnittelu yhteen parametriseen systeemiin. Tämä pienentää suunnittelijan työmäärää sekä parantaa luonnossuunnitelmien laatua. Parametrisointi mahdollistaa myös suunniteltavan rakenteen optimoinnin suunnittelutyön yhteydessä.

1.2 Tutkimuksen tavoitteet

Tämän tutkimuksen tavoitteena on selvittää missä ja miten parametrissa suunnittelua on järkevää hyödyntää rakennesuunnittelussa. Oletetaan, että menettely on järkevää, kun

sillä voidaan saavuttaa joko taloudellista etua tai muunlaista kilpailuetua. Alkuhypoteesina on, että suunnittelemalla parametrisesti suunnittelutyö ainakin joissakin tapauksissa nopeutuu, jolloin saavutetaan taloudellista etua.

Jotta tämä saadaan selville, tulee selvittää mitä parametrinen suunnittelu on, miten sitä voidaan tehdä ja soveltaa erityisesti rakennesuunnittelualalla. Lisäksi tulee selvittää mitä hyötyä saadaan suunniteltaessa kohteita parametrisesti. Jotta tuloksia olisi mahdollista hyödyntää, tulee edelleen selvittää parametrinen suunnittelun sovellettavuutta. Tämä tarkoittaa muun muassa sitä, miten erilaisia suunnittelutehtäviä olisi järkevää parametrisoida, mikä on riittävä mallintamistarkkuus ja miten parametrista suunnitelmaa tai mallia voidaan edelleen hyödyntää asiakkaalle toimitettavassa lopputuotteessa. Luonnollisena jatkumona edellisille toimii optimointi, joka on käytännössä enää matemaattinen tehtävä, kun suunnittelun muut osa-alueet on toteutettu parametrisesti.



Kuva 1.1. Tutkimuksen tavoitteet.

Tutkimuksen tavoitteet ovat toisistaan riippuvaisia ja ne on tehtävä järjestyksessä, joten ne voidaan esittää porraskuvana (kuva 1.1). Kuvassa on esitetty tavoitteet toteuttamisjärjestyksessä. Ensin tulee selvittää mitä parametrinen suunnittelu on, jonka pohjalta voidaan edetä tutkimaan sopivia sovelluskohteita erilaisista rakennesuunnittelutehtävistä. Kun sopivat suunnittelukohteet tunnetaan, voidaan edelleen edetä selvittämään, miten näitä kohteita tulisi parametrisoida. Viimeisenä vaiheena selvitetään, miten parametrista mallia voidaan hyödyntää suunniteltavan rakenteen optimoinnissa.

1.3 Tutkimuksen rajaus

Myös tutkimuksen rajauksessa edetään kuvan 1.1 esittämissä portaissa. Tutkimuksessa käsitellään parametrista suunnittelua kirjallisuusselvityksenä ensin yleisellä tasolla käyden läpi menetelmän teoreettista pohjaa. Käsittely rajataan siten, että se käsittelee parametrista suunnittelua rakennesuunnitteluun soveltuvilta osin. Lisäksi tarkastellaan parametrinen prosessien hyödyntämistä muilla suunnittelualoilla. Kirjallisuusselvityksen perusteella saavutetaan käsitys parametrinen suunnitteluprosessin eduista sekä mahdollisista

ongelmakohdista. Lisäksi päästään selville muista parametrinen prosessin soveltamisen kannalta olennaisista tekijöistä.

Kirjallisuusselvityksen tulosten perusteella rajataan käytännön tarkastelu käsittämään muutamia rakenneosia: palkkia, pilaria, ristikoita ja liitoksia. Edelleen tarkastelu rajautuu rakennusmateriaaleista tarkastelun edetessä pääasiassa teräsrakenteisiin. Yksinkertaisten rakenteiden osalta käsitellään myös puurakenteita. Teräsbetonirakenteiden osalta aihetta tarkastellaan ainoastaan käsitteellisesti. Lisäksi tarkastellaan kokonaisuuden soveltamista yksinkertaisessa teräsrakenteisessa hallirakennuksessa, jossa esiintyy edellä mainittuja rakenneosia. Rakenneosien parametrinen suunnittelun lisäksi työssä tarkastellaan yhteistyömahdollisuuksia arkkitehdin ja rakennesuunnittelijan välillä, kun molemmat osapuolet soveltavat algoritmiavusteisia menetelmiä. Yhteistyötapojen osalta esitellään Australiassa toteutettu esimerkkikohde, jonka suunnittelussa sekä arkkitehti että rakennesuunnittelija ovat yhteistyössä hyödyntäneet parametrisia suunnittelumenetelmiä. Edellisten lisäksi tarkasteltiin liittyviltä osin optimointia kvantitatiivisin menetelmin. Työssä toteutetun ristikkokomponentin avulla luotua ristikkoa optimointiin Rhinoceros-Grasshopper ohjelmiston tarjoamin optimointimenetelmin geneettisellä algoritmilla sekä simuloidulla jäähdytyksellä eri parametrimäärillä.

2. TEORIA

2.1 Mitä on parametrinen suunnittelu

Parametrisen suunnittelun käsite on tällä hetkellä käytössä aloilla, joissa pääasiassa suunnitellaan muotoa. Tällaisia aloja on esimerkiksi arkkitehtuuri ja teollinen muotoilu. Näin ollen parametrisella suunnittelulla tarkoitetaan tyypillisesti sopivan muodon etsimistä käyttäen hyväksi parametrisia, algoritmiavusteisia menetelmiä. Tämä lähestymistapa poikkeaa olennaisesti rakennesuunnittelijan käsityksestä suunnittelusta.

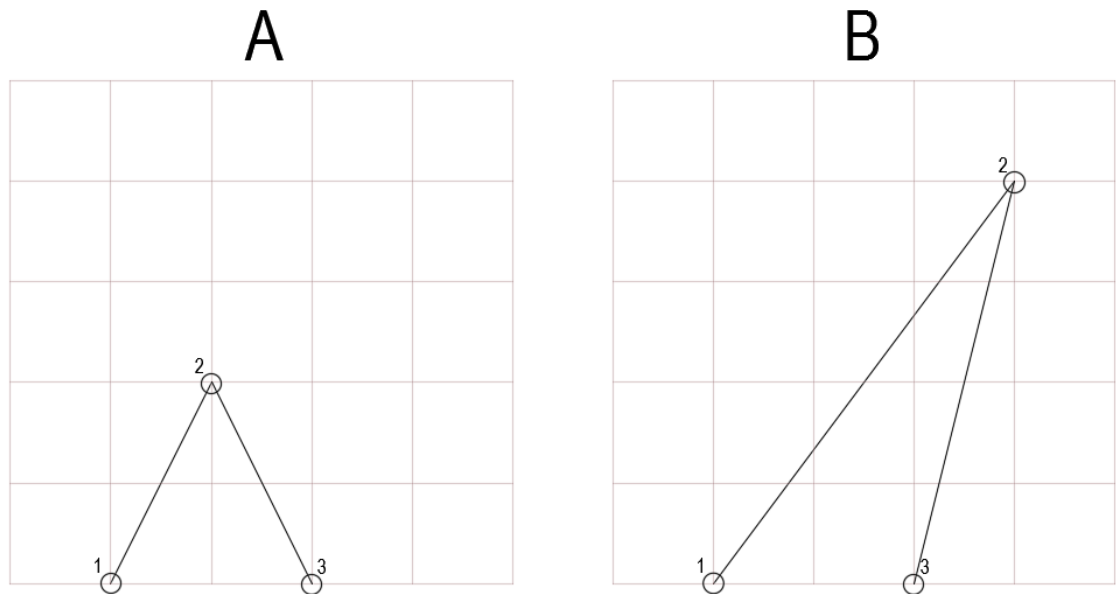
Jotta suunnittelumenetelmä voidaan määritellä parametriseksi, tulee systeemissä olla avoimia parametreja, joiden arvoja muuttamalla voidaan muuttaa suunnitteluratkaisua. Algoritmiavusteisissa suunnittelumenetelmissä (*Algorithms-Aided Design*, AAD) taas vaatimuksena on nimenomaan yksi tai useampi algoritmi, jolla suunnitteluratkaisu tuotetaan. Tässä määritelmässä ei ole vaatimuksia avoimille parametreille, joskin systeemin järkevyyden ilman mahdollisuutta muuttaa parametreja voidaan monissa tapauksissa kyseenalaistaa. Algoritmin yleisen määritelmän perusteella (luku 2.4) voidaan perustella, että parametrinen suunnittelumenetelmä on algoritmiavusteinen suunnittelumenetelmä. Olennaisin ero termien välillä on, mitä menetelmän osa-aluetta halutaan korostaa: parametrisuus korostaa muutosmahdollisuutta parametrien avulla, kun taas algoritmiavusteisuus korostaa suunnitelman tuottamista algoritmien avulla.

Rakennesuunnittelijan pääasiallinen tehtävä on suunnitella rakenteita, jotka ovat turvallisia ja terveellisiä. Tämä tehdään yleensä arkkitehdin suunnitteleman geometrian rajoissa. Näin ollen yleisessä tapauksessa puhuttaessa rakennesuunnittelusta, ei voida puhua parametrisestä suunnittelusta, mikäli ainoastaan luodaan geometrialtaan parametrinen malli, joka vastaa arkkitehdin suunnitelmaa. Tällöin on puhuttava parametrisestä mallintamisesta (Tanska & Österlund 2014). Mikäli tehdään parametrista rakennesuunnittelua, tulee mallin avulla suunnitella itse rakennetta, joka tosin monesti voi tarkoittaa esimerkiksi rakenteen geometrisia ominaisuuksia, kuten palkkikokoja tai ristikon topologiaa.

Yleisesti parametrista suunnittelua voidaan määritellä sen perusperiaatteiden kautta. Sutherland (1964) käsittelee väitöskirjassaan ensimmäistä (Woodbury 2010) koneavusteista ohjelmaa Sketchpadiä. Väitöskirjassaan Sutherland (1964) esittää muutamia perustavia ominaisuuksia joita parametriselta suunnittelumenetelmältä tulee löytyä.

Parametrinen suunnittelu aloitetaan joukolla muuttujia, jotka tyypillisesti voivat olla esimerkiksi geometriakoordinaatteja tai kuormitustietoja. Nämä muuttuja sidotaan toisiinsa reunaehdoilla, kuten vaatimuksella siitä, että kahden viivan päätepisteiden tulee aina sijaita samassa pisteessä. Nämä yhdistämällä voidaan koota yksinkertainen parametrinen

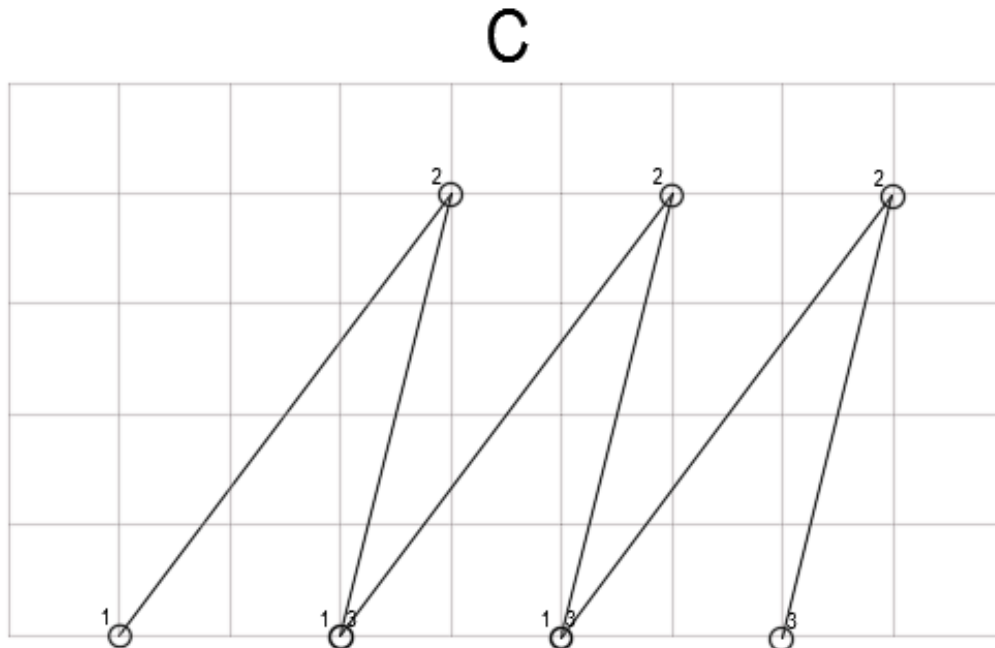
systeemi, jossa on rajattu määrä muuttujia eli parametreja, joita muuttamalla saadaan erilaisia lopputuloksia. (Sutherland 1964) Kuvassa 2.1 on esitetty esimerkki tällaisesta systeemistä.



Kuva 2.1. Esimerkki parametrisesta systeemistä.

Kuvassa 2.1 on yksinkertainen parametrinen systeemi. Muuttujina systeemissä viivojen päiden koordinaatit. Reunaehtoina on vastaavasti, että päiden 1 ja 3 koordinaattien on pysyttävä paikoillaan ja että viivojen on oltava suoria. Näin ollen parametriksi jää pisteen 2 koordinaatti. Tätä parametria muuttamalla voidaan lähtötilanteesta A päästä esimerkiksi lopputulokseen B.

Lisäämällä systeemiin lukumääräparametri, jolla alkugeometriaa voidaan monistaa ja lisäämällä reunaehto, joka määrää että, monistetun geometriaosan pisteen 1 on sijaittava edellisen osan pisteessä 3 voidaan päästä esimerkiksi kuvan 2.2 lopputulokseen C.



Kuva 2.2. Jatkoa kuvalle 2.1.

Kuvassa 2.2 siis esitetyn lukumääräparametrin arvo on kolme, jolloin muodostuu kuuden viivan ryhmä.

Esitetyistä esimerkeistä havaitaan parametrisen systeemin perusluonne, jossa systeemin lopputulosta voidaan muokata parametreja manipuloimalla. Edelleen voidaan määritellä parametrinen suunnittelu menetelmäksi, jossa rakennetaan parametrinen systeemi suunnittelutehtävän reunaehtojen perusteella ja etsitään sopivaa suunnitteluratkaisua manipuloimalla jäljellejääviä parametreja. Rakennesuunnittelussa osa reunaehdoista on tyypillisesti arkkitehdin suunnitelmissaan esittämää geometriaa.

2.2 Parametrit ja reunaehdot

Kuvissa 2.1 ja 2.2 esitetty esimerkki parametrisestä systeemistä kuvaa geometrinen reunaehdotehtävää. Hoffmann & Joan-Arinyo (2005) esittävät, että geometrinen reunaehdotehtävä voidaan määritellä seuraavien joukkojen avulla:

1. E on geometrinen suunnitteluavaruus, johon tehtävä on sijoitettu. E on yleensä euklidinen avaruus, jonka ulottuvuus riippuu ongelmasta. Kuvan 2.1 esimerkissä tehtävä on sijoitettu kaksiulotteiseen euklidiseen avaruuteen R^2 .
2. O tarkoittaa joukkoa geometrisia elementtejä, kuten viivoja, kappaleita tai pisteitä, joiden avulla tehtävä määritellään. Kuvan 2.1 esimerkissä geometriset elementit ovat viivat pisteiden 1-2 ja 2-3 välillä. Elementtijoukkoa voidaan merkitä kaavalla (1)

$$O = \{o_1, o_2, \dots, o_n\}, n \in \mathbb{N}, \quad (1)$$

missä o tarkoittaa yksittäistä elementtiä (joukon alkia) ja n viimeisen elementin yksilöintinumeroa.

3. X tarkoittaa joukkoa tehtävän muuttujia, joiden arvot halutaan määrittää tehtävän ratkaisemiseksi. Näillä muuttujilla kuvataan yleensä jotain geometrista arvoa, kuten pisteen koordinaattia, viivan pituutta tai ympyrän sädettä. Kuvan 2.2 esimerkissä muuttujana on pisteen 2 koordinaatti. Muuttujajoukkoa voidaan merkitä kaavalla (2)

$$X = \{x_1, x_2, \dots, x_m\}, m = 0 \vee m \in \mathbb{N}, \quad (2)$$

missä x tarkoittaa yksittäistä muuttujaa (joukon alkia) ja m viimeisen muuttujan yksilöintinumeroa.

4. C tarkoittaa reunaehtojen joukkoa. Reunaehdot voivat olla joko geometrisia tai yhtälöin määriteltyjä. Geometriset reunaehdot kuvaavat vastaavia asioita kuin muuttujat X sillä erotuksella, että reunaehdon arvo on sidottu esimerkiksi vakioon lukuarvoon tai suoraan jonkin muuttujan arvosta riippuvaksi. Yhtälöin voidaan määritellä matemaattisia riippuvuuksia reunaehtojen ja muuttujien välillä. Kuvan 2.1 esimerkissä geometrisena reunaehtona on pisteiden 1 ja 3 sijainti. Esimerkissä ei ole yhtälöin määrättyjä reunaehtoja. Reunaehtojoukkoa voidaan merkitä kaavalla (3)

$$C = \{c_1, c_2, \dots, c_l\}, l = 0 \vee l \in \mathbb{N}, \quad (3)$$

missä c tarkoittaa yksittäistä reunaehto (joukon alkia) ja l viimeisen reunaehdon yksilöintinumeroa.

Joukkojen luonnetta tarkastellessa huomataan, että suunnitteluavaruuden E on oltava olemassa tehtävän määrittelemiseksi. Avaruuden ulottuvuuden määrittely rajaa muiden joukkoja, koska ne määritellään tässä avaruudessa. Esimerkiksi 2-ulotteisessa avaruudessa ei voida määritellä geometrisiksi elementeiksi tilavuuskappaleita. Elementtijoukko O ei myöskään voi olla tyhjä joukko, vaan sen on koostuttava vähintään yhdestä geometrista elementistä, jotta geometrinen reunaehtotehtävä voi olla olemassa. Muuttujajoukko X voi triviaalisti olla myös tyhjä joukko ($m = 0$) (Hoffmann & Joan-Arinyo 2005), jolloin esitetty tehtävä itsessään joko on tai ei ole ratkaisu. Myöskään reunaehtojen joukko C ei voi olla tyhjä joukko, sillä geometrian tulee vähintään olla sidottuna johonkin määritellyn avaruuden pisteeseen. Kuten edellä huomattiin, yhtälöin määrättyjen reunaehtojen joukko, joka on reunaehtojen C osajoukko, voi olla tyhjä joukko ($l = 0$) (Hoffmann & Joan-Arinyo 2005)

Kun tunnetaan muuttujat, voidaan geometrinen reunaehtotehtävä muodostaa seuraavasti:

- Kun tunnetaan joukot E , O , X ja C , onko elementeille O olemassa sellaista sijaintia avaruudessa E , että reunaehdot C täyttyvät?
- Jos tällainen ratkaisu/ratkaisuja on olemassa, onko olemassa sellainen ratkaisu, jossa muuttujien arvolla X reunaehdot C täyttyvät?

Olennaista on huomata, että osa reunaehdoista voi riippua muuttujista X , jolloin tehtävän reunaehdot mahdollisesti muuttuvat, kun muuttujat tuodaan mukaan systeemiin. (Hoffmann & Joan-Arinyo 2005) Kuvan 2.1 esimerkissä siis kysytään, voidaanko solmujen väliset viivat sijoittaa siten, että niiden toiset päät on sidottu pisteisiin 1 ja 3 ja toiset päät yhdistyvät jossakin pisteessä 2, joka siis voi olla mikä tahansa sijainti määritellyssä avaruudessa. Tässä tapauksessa vastaus on hieman triviaali johtuen jo siitä, että eräs ratkaisu on esitetty jo kuvassa. Edelleen voidaan kysyä, mikä tulee olla pisteen 2 sijainti, jotta reunaehdot täyttyvät. Esimerkin tilanteesta havaitaan nopeasti, että ratkaisuja on ääretön määrä.

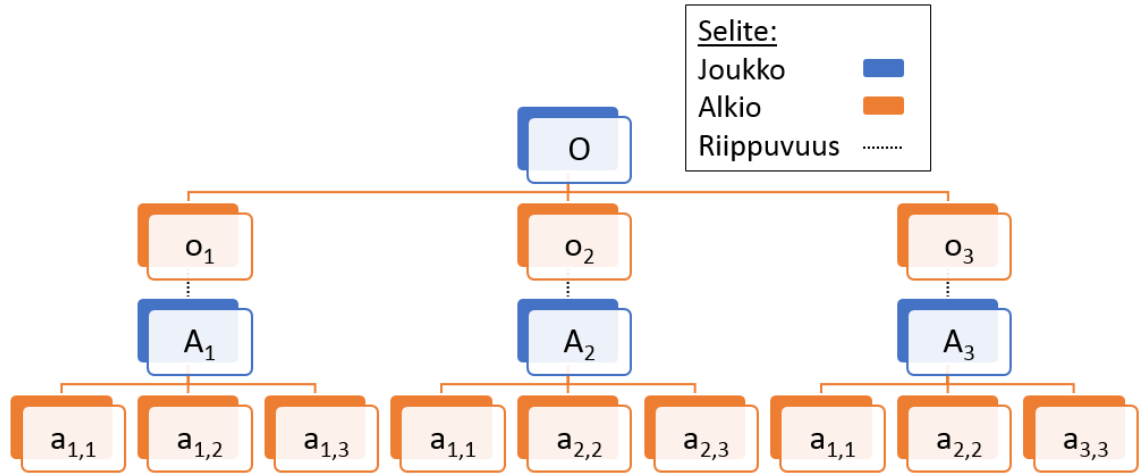
Kuten huomataan, geometrinen reunaehtotehtävä antaa perusteet geometrian parametrisen suunnitteluun. Hoffmannin & Joan-Arinyon (2005) esittämässä tehtävässä elementtjoukko O , muuttujajoukko X ja reunaehtojoukko C liittyvät kaikki tehtävän geometriaan. Kuvassa 2.1 esitetty on tyypillinen esimerkki geometrisesta reunaehtotehtävästä. Tällaisia tehtäviä erilaisilla ratkaisumalleilla sovelletaan laajasti erilaisissa CAD-ohjelmitoissa, joissa esimerkiksi useiden viivojen geometriaa voidaan manipuloida yhteistä pistettä siirtämällä (Bouma et al. 1995).

Yleinen geometrinen reunaehtotehtävä ei kuitenkaan ole riittävän laaja käsittämään rakenteiden parametrisessa suunnittelussa tarvittavia parametreja. Vaikka geometria on olennaisessa osassa rakenteiden suunnittelussa, liittyy rakenteiden suunnittelutehtäviin monia muita tekijöitä, joita esitetyillä joukoilla (E , O , X , C) ei voida kuvata. Tällaisia ovat esimerkiksi materiaaliominaisuudet, kuten lujuus ja kimmokerroin. Lisäksi tehtäviä halutaan usein yksinkertaistaa, ja esimerkiksi kolmiulotteisia sauvoja kuvata yksittäisillä viivoilla. Tällöin viivalle on pystyttävä antamaan myös geometrisia muuttujia, kuten viivan poikkileikkaus, jolloin pystytään edelleen kuvaamaan viivan jäykkyysominaisuuksia. Vaikka poikkileikkaus on geometrinen muuttuja ja se pystyttäisiin kuvaamaan joukon O alkioina, saadaan edellä mainitusta yksinkertaistuksesta hyötyä, koska tällöin reaaliavaruuden tilanne voidaan yksinkertaistaa R^2 avaruuteen. Myös joukon O alkioden lukumäärä pysyy pienempänä mikä edelleen johtaa pienempään joukkoon C .

Näin ollen voidaan määritellä ominaisuusjoukko A_o . Määritellään ominaisuusjoukko A_o kuvaamaan jonkin geometrisen elementin ominaisuuksia. Näin ollen ominaisuusjoukko on riippuvainen elementtjoukon alkioista, muttei kuitenkaan ole elementtjoukon osajoukko. Ominaisuusjoukkoa voidaan merkitä kaavalla (4)

$$A_o = \{a_{o,1}, a_{o,2}, \dots, a_{o,k}\}, k = 0 \vee k \in \mathbb{N}, \quad (4)$$

missä a tarkoittaa yksittäistä ominaisuutta (joukon alkia), o tarkoittaa joukkoa vastaavan elementtijoukon alkion yksilöintinumeroa ja k viimeisen ominaisuuden yksilöintinumeroa. Elementtijoukkoa ja ominaisuusjoukkoa sekä niiden välistä suhdetta on havainnollistettu kuvassa 2.3.



Kuva 2.3. Elementtijoukon ja ominaisuusjoukon välinen yhteys, kun molemmissa joukoissa on kolme alkia.

Kuvassa 2.3 on esitetty elementtijoukko O , johon kuuluu esimerkkitapauksessa kolme alkia (o_1 , o_2 , ja o_3). Ominaisuusjoukot (A_o) ovat riippuvaisia aina niitä vastaavasta elementtijoukon alkioista. Edelleen kuhunkin ominaisuusjoukkoon kuuluu kolme alkia (esimerkiksi $a_{2,1}$, $a_{2,2}$ ja $a_{2,3}$).

Jokaiselle elementille voidaan ominaisuusjoukon avulla määrätä erilaisia ominaisuuksia riippuen elementin geometrisesta luonteesta. Esimerkiksi viivalle voidaan ominaisuuksiksi antaa avaruudessa R^2 määritellyssä tehtävässä (esimerkiksi) poikkileikkaus, materiaali, sauvavoima ja väri. Käytännössä mahdollisuudet ovat rajattomat ja ne riippuvat muodostettavan tehtävän luonteesta ja valitusta sovelluskohteesta.

Tyypillisesti suunnittelutehtävää voidaan rajata, vähintään siten että tunnetaan rakenteeseen kohdistuvat rasitukset. Näin ollen myös ominaisuusjoukko voidaan jakaa muuttujiin A_o^X ja reunaehtoihin A_o^C . Osajoukkoja voidaan merkitä kaavoilla (5) ja (6)

$$A_o^X = \{a_{o,1}^X, a_{o,2}^X, \dots, a_{o,j}^X\}, j = 0 \vee j \in \mathbb{N}, \quad (5)$$

$$A_o^C = \{a_{o,1}^C, a_{o,2}^C, \dots, a_{o,i}^C\}, i = 0 \vee i \in \mathbb{N}, \quad (6)$$

missä a^X tarkoittaa yksittäistä muuttuvaa ominaisuutta (joukon alkia), a^C yksittäistä reunaehto-ominaisuutta o tarkoittaa joukkoa vastaavan elementtijoukon alkion yksilöintinumeroa sekä j ja i viimeisen ominaisuuden yksilöintinumeroa. Koska joukko A_o on jaettu näihin osajoukkoihin, täyttävät ne ehdon

$$A_o = A_o^X \cup A_o^C, \quad (7)$$

eli kunkin alkion on kuuluttava toiseen osajoukkoon.

Huomataan, että myös joukon A_o alkiot on määritelty jossakin geometrisessä avaruudessa R . Avaruus R on matemaattisesti geometrinen, mutta sen kaikki ulottuvuudet eivät vastaa tyypillisesti sijaintiin liitettäviä ulottuvuuksia, joissa yksikkönä on pituuden yksikkö. Lisäksi ne määrittävät joukon O geometria-alkioille muuttujia ja reunaehtoja. Näin saadaan toinen geometrinen reunaehtotehtävä edellä määretyillä joukoilla. Rakennesuunnitteluun liittyvissä sovelluksissa on huomattava, että haluttu geometrinen muoto on jo ratkaistu ensin esitetyn tehtävän avulla ja ominaisuusjoukon avulla tarkastellaan ainoastaan ratkaisun sovellettavuutta käytännössä.

Molempien tehtävien reunaehdot voidaan määrittää parametreiksi. Tämä tarkoittaa sitä, luodaan systeemi, jolla tehtävän reunaehdot voidaan muuttaa halutulla tavalla, joka edelleen muuttaa tehtävän ratkaisua. Kun ratkaisu tarkoittaa sovelluksissa nimenomaan suunnitteluratkaisua, johon annetut reunaehdot johtavat, mahdollistaa tämä erilaisten suunnitteluratkaisujen etsimisen. Näin ollen menetelmä on erityisen käyttökelpoinen tehtävissä, jossa reunaehdot ei vielä lopullisesti tunneta tai kun reunaehdot halutaan määrätä suunnitteluratkaisun kannalta mahdollisimman otollisiksi.

2.3 Geometrisen reunaehtotehtävän ratkaisumenetelmät

Geometrisen reunaehtotehtävän ratkaisuun on olemassa useita menetelmiä, joita on esitetty laajemmin esimerkiksi lähteissä (Bouma et al. 1995; Hoffmann & Joan-Arinyo 2005). Ratkaisutavan valinta riippuu tehtävänasettelusta sekä halutusta lopputuloksesta ja se tehdään perustuen eri menetelmien ominaisuuksiin.

Koska tehtävät ovat monesti huomattavasti esitettyä esimerkkiä vaikeampia, myös ratkaisumenetelmät ovat monimutkaisia ja ne vaativat tyypillisesti tietokoneavusteisuutta ratkaisujen löytämiseksi ja oikean ratkaisun valitsemiseksi. Tällaiset ratkaisijat toimivat tyypillisesti kaikkien graafisten suunnitteluohjelmistojen taustalla (Hoffmann & Joan-Arinyo 2005).

Parametrinen mallintaminen tarkoittaa käytännössä ensin esitetyn reunaehtotehtävän ratkaisua siten, että reunaehdot on parametrisoitu. Parametrinen suunnittelu edellyttää monissa tapauksissa myös toisen geometrisen reunaehtotehtävän ratkaisua, kuten edellisessä luvussa on esitetty.

Algebrallisissa ratkaisutavoissa tehtävä esitetään epälineaarisenä yhtälöryhmänä. Kun yhtälöryhmä on muodostettu, voidaan tehtävä ratkaista millä tahansa yhtälöryhmän ratkaisumenetelmällä. Algebrallisen menetelmän hyvä puoli on se, että sillä voidaan ratkaista teoriassa kaikki ratkaistavissa olevat menetelmät. Käytännössä kuitenkin ratkaisu rajoittuu käytettävissä olevien yhtälöryhmän ratkaisutapojen rajoituksiin. Algebrallisia

ratkaisutapoja ovat esimerkiksi numeerinen menetelmä, symbolinen menetelmä sekä yhtälöryhmäanalyysi. (Hoffmann & Joan-Arinyo 2005).

Numeerisessa ratkaisutavassa tehtävä muutetaan muotoon, jossa se voidaan esittää yhtälöryhmällä. Muodostettu yhtälöryhmä ratkaistaan numeerisesti jollakin iterointimenetelmällä. (Bouma et al. 1995) Näitä iterointimenetelmiä ovat esimerkiksi Newton-menetelmä ja Newton-Rhapson-menetelmä (Bouma et al. 1995; Hoffmann & Joan-Arinyo 2005). Menetelmän hyviä puolia on, että sillä pystytään ratkaisemaan merkittävästi laajempia tehtäviä kuin muilla menetelmillä. Lisäksi sillä pystytään ratkaisemaan tehtäviä, joissa reunaehdot on enemmän kuin vapausasteiden edellyttämä määrä. Menetelmä soveltuu myös epälineaarisiin tehtäviin. (Bouma et al. 1995) Menetelmän huonona puolena on ratkaisemisen vaatima laskentateho, joka on merkittävästi suurempi kuin muissa ratkaisumenetelmissä. Numeerisella menetelmällä saavutettu ratkaisu on myös aina epätarkka, joskin ratkaisun tarkkuus riippuu vaadituista konvergenssin hyväksymisehdoista.

Graafisissa ratkaisutavoissa tehtävä esitetään geometrisina elementteinä, joiden päätepisteet muodostavat elementit ja joiden sijaintia reunaehdot, päätepisteiden väliset viivat, rajaavat. Annettu tehtävä pilkotaan mahdollisimman pieniin aliongelmiksi, joiden ratkaiseminen tunnetaan yksinkertaiseksi. (Hoffmann & Joan-Arinyo 2005) Menetelmän etuna on ratkaisun visuaalisuus, jolloin tehtävästä voidaan visuaalisesti valita haluttu ratkaisu, vaikka reunaehdot toteuttavia ratkaisuja olisi useita. Graafisia ratkaisutapoja ovat rakentava ratkaisutapa, vapausasteanalyysi sekä laajennusmenetelmät (Hoffmann & Joan-Arinyo 2005).

Rakentavassa ratkaisutavassa ratkaisu etsitään liittämällä geometrisia elementtejä, kuten viivoja ja ympyröitä toisiinsa yksi toisensa jälkeen, kunnes ratkaisu löydetään. Nämä elementit määräytyvät reunaehtojen perusteella. (Hoffmann & Joan-Arinyo 2005) Menetelmän hyvä puoli on ymmärrettävyys käyttäjän näkökulmasta: on visuaalisesti helppo ymmärtää, jos määrätty geometriakomponentti eivät sovi toisiinsa tai miten niitä on mahdollista muuttaa, jotta ratkaisu olisi olemassa (Bouma et al. 1995). Toisaalta menetelmän puutteena on, että se soveltuu ainoastaan geometrian ratkaisemiseen, eikä siihen voida lisätä muita, ulkoisia muuttujia (Hoffmann & Joan-Arinyo 2005).

Edellä esitetyt ratkaisumenetelmät ovat matemaattisesti haastavia, jolloin niiden hyödyntäminen erilaisilla parametreilla laajennetussa geometrisessa reunaehtotehtävässä on hankalaa tai joissain tapauksissa mahdotonta. Lisäksi hankaloittavana tekijänä on se, ettei geometrian reunaehtojen ja laajennettujen reunaehtojen välisiä riippuvuuksia tunneta tai niitä ei osata lausua matemaattisesti. Aiheesta ei ole myöskään julkaistu tutkimusaineistoa, jonka pohjalta työtä olisi mahdollista lähteä tekemään. Näin ollen se edellyttää sopivien ratkaisumenetelmien ja sovellustapojen tutkimista, mikä vaatii myös hyvää matemaattista osaamista. Koska käytännön suunnittelutyössä tehtyjen sovellusten hyödyntäminen uudestaan on rajallista, on myös tällaisen selvityksen tekeminen kyseenalaista, mikäli ongelma on ratkaistavissa muulla menetelmällä.

Rakennesuunnittelussa tarkastellaan tyypillisesti tarkasteltavat suunnittelutehtävät, kuten rakenteiden kuormankantokapasiteetti, voidaan jakaa äärelliseen määrään diskreettejä tarkastelutapauksia, kuten esimerkiksi palkkipoikkileikkaukset. Tämän tiedon avulla tehtävä voidaan jakaa kahteen osaan: geometrian ratkaisuun geometrisella reunaehtotehtävällä ja lisäehtojen tarkistamiseen. Lisäehtojen tarkistaminen voidaan tehdä hyödyntämällä algoritmia, joka laskee kaikki tai määrättyltä väliltä diskreetit tapaukset läpi ja vertaa niitä annettuihin reunaehtoihin. Algoritmilla siis tarkistetaan numeerisesti laajennettujen reunaehtojen täyttyminen ilman, että välttämättä tiedetään miten reunaehdot riippuvat geometriasta. Huomattavaa on, että tämä menetelmä soveltuu ainoastaan diskreetteihin tai diskretisoituihin tapauksiin eikä sillä voida tutkia jatkuvia tapauksia. Jatkuva tapaus edellyttää geometrisen reunaehtotehtävän täydellistä ratkaisua.

2.4 Algoritmit

Algoritmi on äärellinen joukko ohjeita, joita seuraamalla voidaan suorittaa määrätty tehtävä. Lisäksi algoritmin tulee toteuttaa seuraavat ehdot:

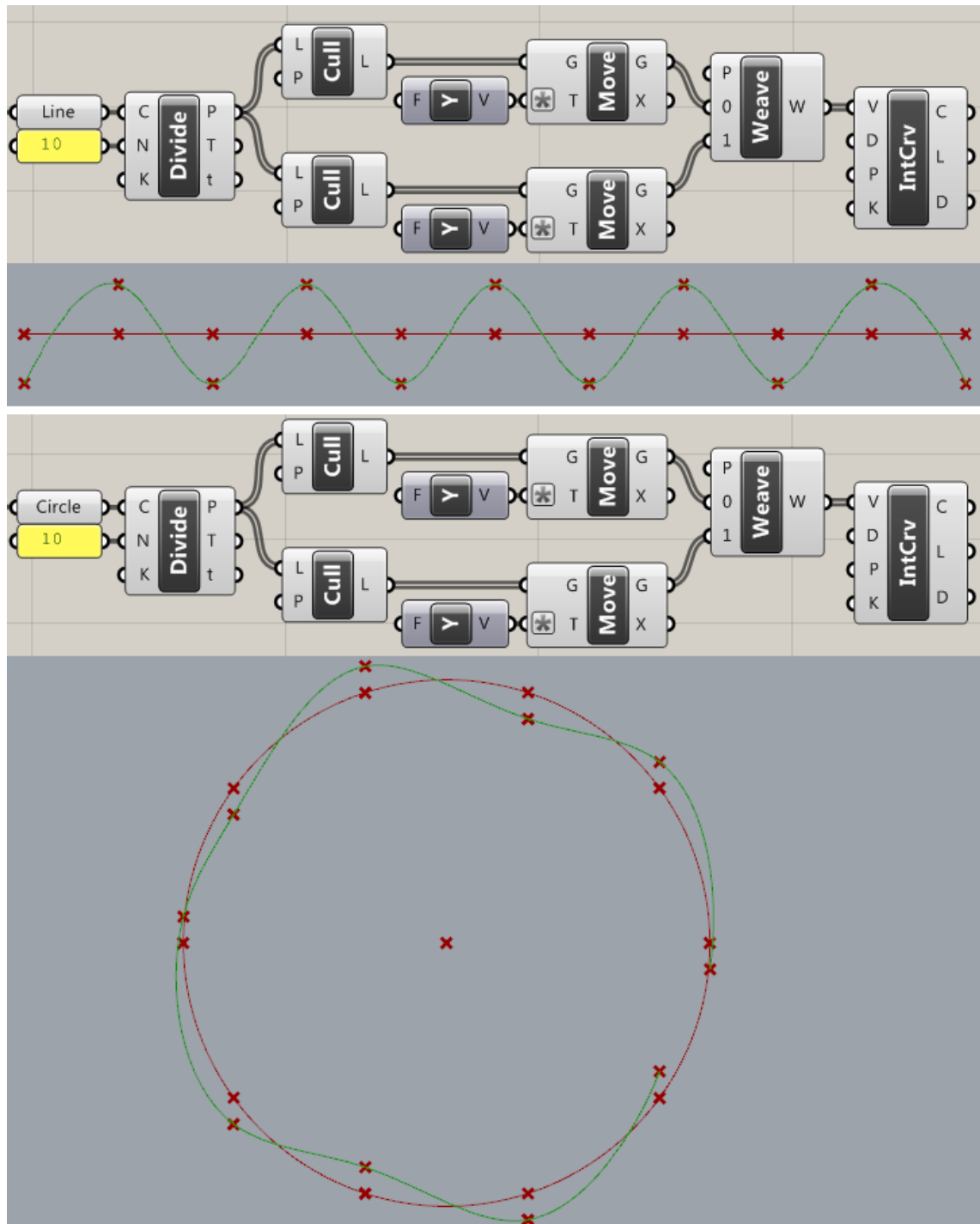
1. Algoritmille tulee antaa ulkoinen **syöte**.
2. Algoritmilla pitää olla **tulos**.
3. Jokaisen ohjeen tulee olla **yksiselitteinen**.
4. Algoritmin tulee olla **päättävä**.
5. Jokaisen ohjeen on oltava **toteuttamiskelpoinen**. (Horowitz & Sahni 1978)

Ehtojen täyttämisestä seuraa, että algoritmi alkaa syötteellä. Se voi olla algoritmin luonteesta riippuen mitä tahansa, esimerkiksi numero tai geometriaelementti. Syöte voi olla myös tyhjä, mutta myös siinä tapauksessa se tulee antaa algoritmille (Horowitz & Sahni 1978). Syötteen antamisen jälkeen suoritetaan yksi tai useampia tehtäviä annetun ohjeen perusteella. Yksittäinen tehtävä on diskreetti ja se on aina suoritettava ennen seuraavaa tehtävää yksiselitteisyyden säilyttämiseksi. Näitä tehtäviä on oltava äärellinen määrä, algoritmi ei voi olla päättymätön. Tähän liittyen yleensä oletetaan, että algoritmin on oltava päättävä sopivaksi määrättyllä aikavälillä (Horowitz & Sahni 1978). Tämä tarkoittaa, että vaikka algoritmi olisi teoriassa päättävä, mutta sen suorittamiseen kuluva aika olisi esimerkiksi miljoonia vuosia toteuttamishetken teknologialla, ei algoritmin voida katsoa olevan päättävä (Horowitz & Sahni 1978). Sopiva aikaväli on aina määriteltävä sovelluskohtaisesti. Toteuttamiskelpoisuuden ehto sulkee pois ilmiselvästi toteuttamiskelvottomat tehtävät, kuten suoran $y = 1$ nollakohtien ratkaiseminen avaruudessa R^2 . Kuitenkin myös teoriassa mahdollinen tehtävä, kahden irrationaaliluvun summa, katsotaan toteuttamiskelvottomaksi (Horowitz & Sahni 1978), koska ilman pyöristämistä luvuissa on ääretön määrä desimaaleja, joiden yhteenlasku taas johtaisi algoritmin päättymättömyyteen. Algoritmin suorittamisen jälkeen saadaan tulos, joka riippuu algoritmin tehtävien määrämällä tavalla syötteestä. Toisaalta voidaan sanoa algoritmin luonteesta johtuen, että sen tulos on aina sama riippumatta suorituskerrasta tai muusta ulkoisesta tekijästä, ellei syötettä tai itse algoritmia muuteta.

Algoritmeja muodostetaan, jotta syöttötietoja voidaan automatisoidusti muokata halutuksi lopputuloksiksi. Koska tuloksen yleensä halutaan täyttävän joitakin ehtoja, tulee ohjeiden olla oikein muodostettuja (Tedeschi 2014). Näin ollen voidaan todeta, että vaikka vaatimus ohjeiden yksiselitteisyydestä täyttää algoritmin reunaehdot, se ei takaa, että lopputulos vastaa haluttua.

Esitettyjen vaatimusten perusteella algoritmit soveltuvat hyvin tietokoneympäristöön, jossa yksikertaisia tehtäviä pystytään suorittamaan tehokkaasti suuria määriä. Tietokoneita käytettäessä ainoana vaatimuksena on, että lähtötiedot, ohjeet ja lopputulos, eli ongelma, voidaan kirjoittaa matemaattisessa muodossa. Tekniset asiat kuten monimutkaisuus tai pituus eivät ole ongelmia, kunhan esitetyt vaatimukset täytetään. Algoritmit kirjoitetaan yleensä ohjelmointikielellä, mikä takaa ohjeiden eli komentojen yksiselitteisyyden (Woodbury 2010). Ongelman lausuminen matemaattisesti ei geometrian tapauksessa tai yleensäkin suurimmassa osassa insinööritehtäviä tuota vaikeuksia, koska ne ovat usein jo valmiiksi matemaattisia tehtäviä.

Käytännössä algoritmien muodostaminen ohjelmointikielellä vaatii kuitenkin tietoteknistä osaamista. Siksi useat ohjelmistovalmistajat ovat kehittäneet visuaalisia ohjelmointityökaluja, joiden avulla ohjelmointia osaamaton käyttäjä voi muodostaa algoritmeja esimerkiksi geometrian muodostamiseen. Yksi visuaalisen ohjelmoinnin muoto ovat solmupohjaiset kaaviot. Tällaiset ohjelmat antavat käyttäjän muodostaa prosessikaavioita, joiden solmuissa toteutetaan jokin toimenpide, algoritmin tehtävä. Ohjelmistolla muodostettavalla visuaalisella algoritmilla voidaan edelleen hallita ohjelmiston muodostamaa geometriaa. Kuvassa 2.4 on esitetty yksinkertainen esimerkki visuaalisesta algoritmista.



Kuva 2.4. Esimerkki visuaalisesta algoritmista.

Kuvan algoritmissa annettu käyrä, esimerkin tapauksessa jana tai ympyrä, jaetaan 10 osaan. Osien päätepisteistä joka toista siirretään positiiviseen y-suuntaan ja joka toista negatiiviseen y-suuntaan vakiomäärä. Tämän jälkeen viivojen kautta piirretään interpolaatiokäyrä.

Parametrisessa suunnittelussa algoritmeja tarvitaan sekä reunaehtojen että suunnittelutehtävän muodostamiseen. Myös itse suunnittelutehtävä on mahdollisesti tarpeellista ratkaista algoritmisella ratkaisumenetelmällä.

2.5 Parametrisen mallin parametrimäärä

Parametrasta mallia luotaessa tulee valita tarkasteltavien parametrien määrä. Periaatteessa kaikki malliin vaikuttavat muuttujat ovat lähtökohtaisesti parametreja. Käytännössä kuitenkin suunnitteluavaruutta on syytä rajoittaa tehtävän tarkastelua varten, jolloin tehtävää luotaessa intuitiivisesti rajataan suunnitteluavaruutta eli lukitaan parametreja.

Kun tehtävää on intuitiivisesti rajattu siihen asti, että tiedossa on käytettävä rakenneratkaisu, tulee valita tämän rakenneratkaisun kannalta olennaiset parametrit. Parametrien valinta voidaan tehdä väliltä, jonka toisessa päässä on kaikkien rakenneratkaisun ominaisuuksiin vaikuttavien parametrien jättäminen auki. Käytännössä tämä tarkoittaa työlästä mallin luontia (Davis 2013), koska tällaisten parametrien määrä on erittäin suuri (ks. kuva 4.8). Toinen ääripää on vain yhden parametrin auki jättäminen. Jos yhtään parametria ei jätetä auki, ei voida puhua parametriseistä suunnittelusta. Davis (2013) ilmaisee asian siten, että ideaalisella parametriseillä mallilla voidaan käsitellä kaikki toivottavat suunnitteluratkaisut mahdollisimman pienellä parametrimäärällä.

Parametrien valinnan perusteena on parametrisein mallin käyttötarkoitus. Näin ollen valinta tulee tehdä tietoisesti mallin luontivaiheessa mallin tarkoitusta ajatellen. Valintaan vaikuttavat

1. rakenteellisten ominaisuuksien varmistaminen
2. haluttu geometrian muokattavuuden taso
3. optimointitarve
4. tulosten hyödyntäminen jatkokäsittelyssä
5. mallin tai sen osien mahdollinen hyödyntäminen tulevaisuudessa.

Tärkein parametrien valintaan vaikuttava kriteeri on rakenteellisten ominaisuuksien varmistaminen. Kun tehdään parametriseistä rakennesuunnittelua, on parametrit valittava siten, että käypä rakenneratkaisu on saavutettavissa. Yhtenä syynä parametriseille mallintamiselle on geometrian helppo muokattavuus. Jotta tämä voidaan saavuttaa, tulee parametrit valita siten, että geometria on muokattavissa halutulla tavalla. Mikäli parametriseistä mallia halutaan hyödyntää optimointiin, on parametrit valittava siten että optimointi on mahdollista ja tarkoituksenmukaista.

Myös parametrisein mallin jatkokäsittelyn tarve asettaa vaatimuksia parametrien valinnalle. Parametriseistä muodostettua tietoa voidaan hyödyntää esimerkiksi FEM-ohjelmistoissa, tietomallintamisessa tai kustannuslaskennassa. Tällöin parametrien valinnassa tulee kiinnittää huomiota siihen, että mallilla voi luoda jatkokäsittelyssä tarvittavaa tietoa sekä siihen, että tämä tieto on muokattavissa jatkokäsittelyanalyysin tulosten perusteella. Jatkokäsittelyn huomioon ottaminen parametriseistä mallia luotaessa edellyttää osaamista ja tietoa jatkokäsittelystä ja siihen olennaisesti vaikuttavista tekijöistä.

Viimeinen tärkeä tekijä mallin luomisessa on tulevaisuudessa toistuvien tarpeiden tunnistaminen parametrisessa mallintamisessa. Parametrusten mallien luominen on suhteellisen työläs prosessi verrattuna perinteiseen suunnittelumenetelmään kuten luvussa 4.3 on esitetty. Koska mallin luontia voidaan nopeuttaa hyödyntämällä aiemmin luotuja algoritmeja ja niiden osia, on järkevää luoda uudet mallit siten, että ne ovat mahdollisimman tehokkaasti uudelleenhyödynnettävissä. Tämä tarkoittaa, että mikäli useamman parametrin auki jättäminen parantaa mallin sovellettavuutta tulevaisuudessa, saattaa parametrin auki jättäminen olla järkevää, vaikka se johtaisi suurempaan työmäärään mallin luonnissa. Tämä menettelytapa on todettu tehokkaaksi myös Aberdeen Groupin raportissa (2007).

Toisaalta juuri oikeiden parametrin valitsemisen on havaittu olevan vaikea tehtävä ainakin arkkitehtisuunnittelussa (Smith 2007), missä suunnitteluavaruus on periaatteessa rajaton. Rakennesuunnittelussa suunnitteluavaruus on kuitenkin pienempi johtuen juuri arkkitehdin geometrialle asettamista reunaehdoista. Rakennesuunnittelijan mahdolliset parametrivalinnat onkin rajattu rakenneosakohtaisesti kuten luvussa 4.4 on esitetty.

3. OPTIMOINTI

3.1 Mitä optimointi tarkoittaa

Kantavien rakenteiden suunnittelussa eri rakenneosien pääasiallinen tehtävä on välittää rakennukseen kohdistuvat ulkoiset kuormat perustuksille. Koska rakennesuunnittelijan käytössä on lukuisia määriä vaihtoehtoja erilaisten kuormien siirtämisellä, on olemassa myös laaja joukko ratkaisuja, joilla tämä tehtävä voidaan toteuttaa. Käytettävissä olevia ratkaisuja rajaa yleensä käytännön tapauksissa jollakin tasolla ennalta määrätty geometrinen muoto ja muut rakenneosalle asetut vaatimukset kuten esimerkiksi lämmönjohtavuus. Yleensä käyttökelpoisten ratkaisujen joukko on silti niin laaja, että kaikkien vaihtoehtojen tutkimiseen kuluisi valtava määrä suunnittelutyötä (Arora 2004).

Jos ja kun suunnittelutehtävä voidaan esittää matemaattisesti, voidaan hyvän suunnitteluratkaisun etsimiseen käyttää tietokoneohjelmistoja (Arora 2004). Tällöin tietokoneen voidaan antaa ratkaista suuri, kuitenkin ennalta rajattu, joukko tehtäviä ja valita niistä jollakin kriteerillä paras. Vaikka jo tätä menetelmää voidaan kutsua optimoinniksi, on sillä kuitenkin kolme heikkoutta verrattuna yleensä sovellettaviin optimointimenetelmiin:

1. Ratkaisu riippuu tehtävän matemaattisen muodon lisäksi ratkaisujoukon rajoituksesta
2. Ratkaisujoukon on oltava diskreetti
3. Menetelmä on laskentatehoa kuluttava, koska muodostetun tehtävän matemaattisia ominaisuuksia ei hyödynnetä parhaan ratkaisun sijainnin etsimisessä, vaan kaikki tehtävän variaatiot lasketaan.

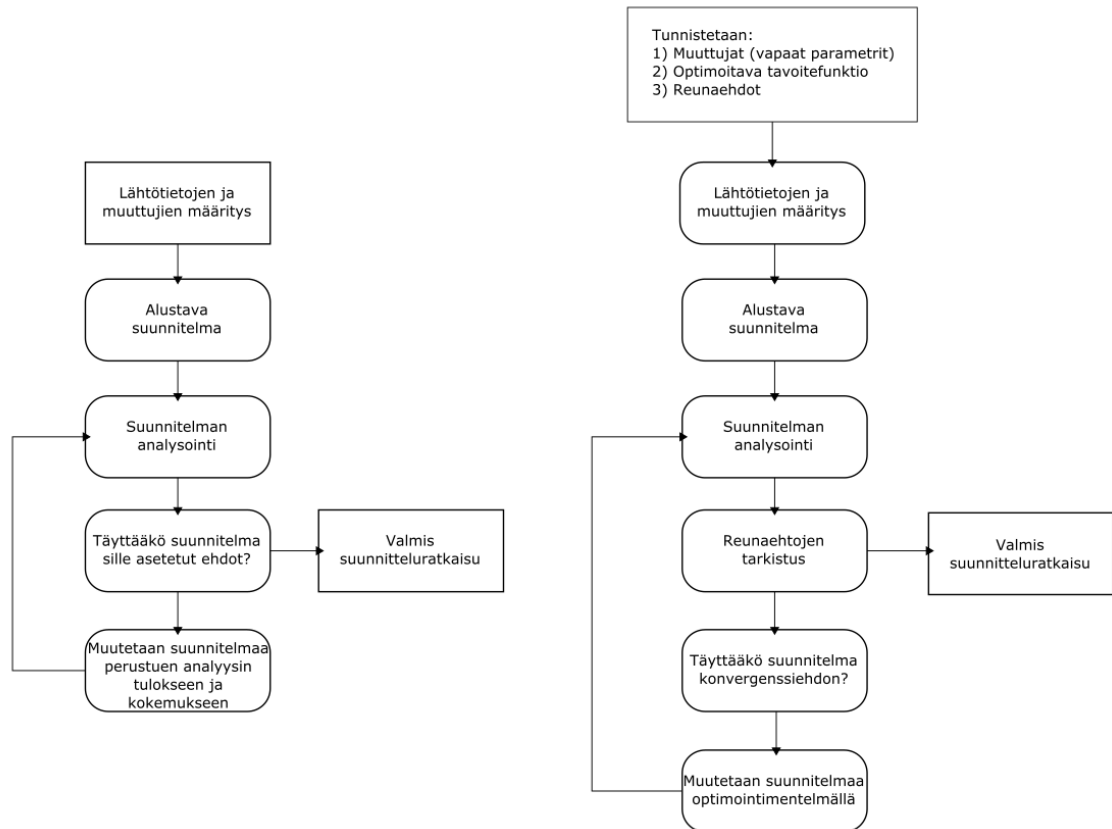
Edellä mainittujen seikkojen vuoksi optimoinnilla tarkoitetaan tyypillisesti optimointialgoritmin avulla suoritettua optimointitehtävää. Erilaisten optimointialgoritmien avulla optimointitehtävän ratkaisua voidaan ohjata kohti tavoitefunktion lokaalia tai globaalia minimiä tarkastelemalla esimerkiksi tavoitefunktion ominaisuuksia. Erilaisia ja eritavoin toimivia optimointialgoritmeja on useita ja niiden sovelluskelpoisuus vaihtelee riippuen tarkasteltavasta optimointitehtävästä.

3.2 Optimointi prosessina

Perinteistä suunnittelumenetelmää voidaan ajatella kokeilun ja erehdyksen menetelmänä. Käytössä oleviin lähtötietoihin perustuen suunnittelija asiantuntemukseensa ja kokemukseensa perustuen tekee arvauksen soveltuvasta suunnitteluratkaisusta. Tämän jälkeen suunnittelija analysoi ratkaisunsa verraten sitä vallitseviin suunnittelukriteereihin, toteaa

soveltuuko ratkaisu tarkoitukseensa sekä havainnoi ratkaisun mahdolliset heikkoudet. Tämän jälkeen suunnittelija joko hyväksyy ratkaisun tai kehittää sitä parhaaksi näkemällään tavalla tekemällä uuden arvauksen suunnitteluratkaisusta.

Optimoinnissa suunnitteluratkaisun löytäminen perustuu tehtävän matemaattiseen formulointiin. Suunnittelutehtävä aloitetaan määrittelemällä muuttujat eli parametrit, optimoitava funktio sekä reunaehdot. Näiden perusteella tehdään jokin alkuarvaus, suoritetaan analyysi, tarkistetaan reunaehdot, tarkistetaan optimointimenetelmän konvergenssiehto ja tarpeen mukaan muutetaan alkuarvausta. Konvergenssiehdolla tarkoitetaan ehtoa, jonka perusteella tarkastellaan, onko tavoitefunktion arvo muuttumassa parempaan suuntaan, mikäli toteutetaan uusi iteraatiokierros (Arora 2004). Tämä voi tarkoittaa esimerkiksi yksinkertaisesti funktion derivaattaa, jonka tulee lokaaleissa ääriarvoissa olla nolla, ja jos arvo poikkeaa nollasta, tiedetään ettei tulos ole ääriarvo. Konvergenssiehdon määrittäminen ja käyttö on tärkeää erityisesti numeerisissa menetelmissä, joissa ääriarvojen sijaintia ei tiedetä, vaan ne joudutaan etsimään. Analyttisissä menetelmissä ääriarvojen sijainnit pystytään yleensä määrittämään analyttisin menetelmin.



Kuva 3.1. Perinteinen suunnitteluprosessi ja optimointisuunnitteluprosessi (muokattu lähteestä Arora 2004).

Kuvassa 3.1 on esitetty perinteinen suunnittelumenetelmä sekä optimointiin perustuva suunnittelumenetelmä prosessikaavioina. Kuvasta huomataan, että perinteisessä suunnittelumenetelmässä suunnitelman kehittyminen perustuu inhimillisiin tekijöihin. Suunnittelija voi hyväksyä suunnitteluratkaisun, kun se täyttää sille asetetut reunaehdot. Toisaalta suunnittelija voi kokemukseensa ja asiantuntemukseensa (sekä käytettävissä olevaan aikaan) perustuen pystyä kehittämään suunnitelmaa kohti parempaa ratkaisua. Toisin sanoen lopputuloksena saatava ratkaisu riippuu merkittävästi sen tekijästä. Optimoinnissa taas ratkaisu riippuu optimointitehtävän määrittelystä sekä valitusta optimointimenetelmästä. Tällöin suunnittelijan tehtäväksi jää näiden tekijöiden sopiva määrittely suhteessa tehtävään.

3.3 Optimointitehtävän formulointi

Optimointitehtävän formulointi on kriittisin ja yleensä työläin vaihe optimoinnissa (Arora 2004). Saatu lopputulos riippuu olennaisesti tehtävän muodostamisesta: muuttujista, optimoitavista funktioista sekä reunaehdosta. Erilaiset virheet ja epätarkkuuden näissä johtavat siihen, ettei optimia tai edes lähellä sitä olevaa ratkaisua voida saavuttaa.

Optimointitehtävän matemaattista muodostamista varten tulee selvittää tai määritellä seuraavat asiat:

- optimointitehtävän tavoite
- lähtötiedot
- muuttujat
- optimointikriteeri
- reunaehdot (Arora 2004).

Optimointitehtävän tavoite antaa reunaehdot muiden edellä mainittujen asioiden määrittämiseksi, koska optimointi tehdään juuri tämän tavoitteen saavuttamiseksi. Lähtötiedoilla tarkoitetaan optimointitehtävän kannalta kriittisiä tietoja, kuten käytettävissä olevia materiaaleja ja niiden hintatietoja. Muuttujat ovat asioita, joita muutamalla suunnittelu-tehtävä tulee ratkaista. Yksinkertaisena esimerkkinä tietyn pituisen suorakaidepalkin optimoinnissa muuttujia voivat poikkileikkauksen ominaisuuksista olla leveys ja korkeus. Muuttujien tulee olla toisistaan riippumattomia. Näin ollen tässä esimerkissä pinta-ala ei käy muuttujaksi, koska se riippuu nimenomaan aiemmin määritellyistä muuttujista. Muuttujien tulee myös olla siinä mielessä oikeita, että tehtävä olisi muodostettavissa. Edelleen mikäli edellisessä esimerkissä tavoitteena olisi optimoida taivutuskapasiteettia, muuttujana ei voitaisi käyttää pinta-alaa, koska palkin taivutusvastus ei riipu siitä. Toisistaan riippumattomien muuttujien määrä määrittelee optimointitehtävän vapausasteen. (Arora 2004)

Jotta eri suunnitteluratkaisut olisivat keskenään vertailukelpoisia, tarvitaan optimointikriteeri. Optimointikriteerin avulla määritellään, kumpi kahdesta muuttujien eri arvoilla muodostetusta ratkaisusta on parempi. Jotta optimointitehtävä voidaan ratkaista, tulee optimointikriteerin olla varioitavien muuttujien skalaarifunktio. Tätä funktiota kutsutaan tavoitefunktioksi, jota minimoimalla tai maksimoimalla löydetään optimointitehtävän ratkaisu. (Arora 2004) Tavoitefunktion määrittäminen riippuu suoraan optimointitehtävän tavoitteesta: jos tavoitteena on saavuttaa mahdollisimman edullinen ratkaisu, tulee määritellä tavoitefunktio, joka kuvaa ratkaisun hintaa. Tällöin muodostettua tavoitefunktioita voidaan minimoida itse optimointitehtävän tavoitteen saavuttamiseksi.

Jotta optimointitehtävä olisi täysin määritelty, tulee vielä määritellä reunaehdot ja niiden riippuvuussuhteet. Reunaehdot riippuvat tehtävän luonteesta: tyypillisissä rakennustekniikan sovelluksissa rakenneosan kuormituskapasiteetti on olennainen reunaehto. Esimerkiksi saatavilla olevien materiaalien ja käytettävissä olevan tilan perusteella voidaan muodostaa tehtävän reunaehdot. Reunaehdot voidaan jakaa neljään eri luokkaan:

- lineaariset ja epälineaariset reunaehdot
- toteutuskelpoisuuden määrittävät reunaehdot
- yhtäsuuruutta ja epäyhtäsuuruutta kuvaavat reunaehdot
- implisiittiset reunaehdot (Arora 2004).

Lineaariset ja epälineaariset reunaehdot määrittävät muuttujien välisiä riippuvuuksia matemaattisten, joko lineaaristen tai epälineaaristen funktioiden avulla. Toteutuskelpoisuutta määrittävillä reunaehdoilla määrätään, että tuloksena saatava suunnitteluratkaisu on toteutuskelpoinen. Yksinkertaisimmillaan tämä tarkoittaa sitä, etteivät esimerkiksi rakenneosan mittoja kuvaavat muuttujat voi saada negatiivisia arvoja. Yhtäsuuruutta ja epäyhtäsuuruutta kuvaavilla reunaehdoilla muuttujan arvot määrätään yhtä suuriksi tai esimerkiksi pienemmäksi kuin jokin vakio, muuttuja tai funktio. Implisiittisillä reunaehdoilla tarkoitetaan sellaisia reunaehtoja, joilla määrätään asioita, joiden riippuvuutta määrittelyistä muuttujista ei suoraan tunneta. Tämä tarkoittaa sitä, ettei reunaehto voida muodostaa edellä mainituilla tavoilla vaan se riippuu tuntemattomalla tavalla muuttujista ja joko toteutuu tai ei toteudu niiden eri arvoilla. (Arora 2004)

Edellisten perusteella voidaan määrittää optimointitehtävän standardimalli. Standardimallissa optimointitehtävä muokataan muotoon, jossa minimoidaan tavoitefunktio. Epäyhtäsuuruutta kuvaavat reunaehdot muokataan sellaiseen muotoon, jossa muuttujan arvo on aina pienempi tai yhtä suuri kuin 0. Tällöin standardimalli voidaan esittää seuraavasti: "Etsitään vektori $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n)$, joka toteuttaa yhtäsuuruusreunaehdot

$$h_j(x) = h_j(x_1, x_2, \dots, x_n) = 0; \quad j = 1 \dots p \quad (8)$$

sekä epäyhtäsuuruusreunaehdot

$$g_i(x) = g_i(x_1, x_2, \dots, x_n) \leq 0; \quad i = 1 \dots m \quad (9)$$

siten että tavoitefunktio

$$f(x) = f(x_1, x_2, \dots, x_n) \quad (10)$$

saa pienimmän mahdollisen arvonsa. (Arora 2004) Tulee huomata, että yhtäsuuruusehtojen lukumäärän p tulee olla pienempi tai yhtä suuri kuin epäyhtäsuuruusehtojen määrän m . Jos $p > m$, osa yhtäsuuruusehdoista on lineaarisesti riippuvia tai yksinkertaisesti vääriä. Jos $p = m$, olemassa on vain yksi ratkaisu, eikä optimointi näin ollen ole tarpeellista. (Arora 2004)

Optimointitehtävän muuttujat (edellä x_n) voivat olla määrittelyltään olla joko diskreettejä tai jatkuvia. Pelkästään jatkuvilla muuttujilla määriteltyä optimointitehtävää kutsutaan jatkuvaksi optimointitehtäväksi. Vastaavasti pelkästään diskreeteillä muuttujilla määriteltyä optimointitehtävää kutsutaan diskreetiksi optimointitehtäväksi. Tehtävä, jonka määrittelyssä käytetään molempia muuttujatyyppejä, on luonteeltaan sekalukuongelma. Jatkuva optimointitehtävä on tyypillisesti edellä mainituista helpoin ratkaista. (Jalkanen 2007) Tämä johtuu siitä, että jatkuvien funktioiden käsittelyssä voidaan käyttää apuna yksinkertaisia matemaattisia menetelmiä, kuten derivointia, apuna ääriarvojen etsintään. Koska diskreettien tai sekalukuongelmien tavoitefunktio ei ole jatkuva, eikä näin ollen derivoituva, ei näitä funktion ominaisuuksia voida hyödyntää optimoinnissa.

Mikäli funktion ominaisuuksia ei voida hyödyntää optimointitehtävän ratkaisussa, käytössä on erilaisia heuristiikkaan perustuvia optimointimenetelmiä. Tällaisia ovat paikallisen haun algoritmit sekä populaatioperusteiset algoritmit (Jalkanen 2007). Paikallisen haun menetelmiä ovat esimerkiksi simuloitu jäähdytys (*simulated annealing*, SA) ja tabuetsintä (*tabu search*, TS). Populaatioperusteisia algoritmeja ovat taas esimerkiksi geneettisiin algoritmeihin (*genetic algorithm*, GA) perustuvat menetelmät sekä parveilualgoritmi (*particle swarm optimization*, PSO). Heuristisille menetelmille on tyypillistä, että yhteen optimointitehtävään soveltuva menetelmä ei ole soveltuva toiseen (Kirkpatrick et al. 1987). Lisäksi on huomattava, että heuristiset menetelmät perustuvat tavoitefunktion arvon määrittelyyn valitun menetelmän määäämissä pisteissä pyrkien algoritmien avulla etenemään kohti parempaa ratkaisua. Näin ollen näillä menetelmillä ei voida todeta ratkaisun luonteesta (lokaali vai globaali optimi) mitään. Edellä mainittuja menetelmiä on onnistuneesti sovellettu rakenteiden optimoinnissa diskreeteissä ja sekalukutehtävissä (Jalkanen 2007).

Simuloidun jäähdytyksen menetelmä on peräisin Metropolis et al. (1953) tutkimuksesta, jossa aineen (partikkelien välistä) tasapainotilaa tietyssä lämpötilassa etsittiin las kennallisesti tutkimalla partikkelien sijainnin muutoksesta aiheutuvaa potentiaalienergian muutosta. Prosessissa jollekin äärellisen partikkelijoukon partikkelien sijaintia muuttamalla tarkasteltiin muuttuuko systeemin potentiaalienergia. Mikäli potentiaalienergian muutos on negatiivinen, muutos sallitaan. Mikäli se taas on positiivinen, muutos sallitaan todennäköisyyden perusteella. Tarkasteluun otetaan satunnaisluku välillä $0 \dots 1$ ja jos se on pienempi kuin $\exp(-\Delta E/kT)$, missä ΔE on potentiaalienergian muutos, k on vakiokerroin ja T lämpötila, muutos hyväksytään ja otetaan edelleen uudeksi lähtötilanteeksi. Ratkaisu saadaan, kun systeemissä ei enää tapahdu hyväksyttäviä muutoksia. (Metropolis et al. 1953)

Asettamalla energian paikalle tavoitefunktio, käyttämällä partikkelien asemesta optimoitavia muuttujia ja käyttämällä lämpötilaa kontrolliparametrina, voidaan menetelmää soveltaa optimointiin. Simulointi aloitetaan korkeasta lämpötilasta, joka sallii paljon ja suuriakin muuttujien välisiä muutoksia (lämpötila on eksponenttifunktion muuttujan nimitäjässä, jolloin muuttujan arvot ovat suuria, eksponenttifunktion arvo on suuri ja edelleen todennäköisyys alittaa tämä arvo satunnaisluvulla on suuri). Kun saavutetaan staattinen tila, lämpötilaa jäähdytetään askel askeleelta, kunnes saavutetaan ”jäätyminen” eli niin pieni lämpötila, ettei muutoksia enää tapahdu. (Kirkpatrick et al. 1987)

Tabu-etsinnällä tarkoitetaan menetelmää, jossa optimiratkaisua etsitään sen hetkisen ratkaisun läheltä. Muuttujien \mathbf{x} mahdolliset muutokset kuuluvat joukkoon $S(\mathbf{x}) - T$, missä $S(\mathbf{x})$ on kaikkien mahdollisten muutosten joukko ja T on tabujoukko, joka tarkoittaa sitä muutosjoukkoa, jota ei sallita. Iteraatiokierroksilla valitaan siirtymä s_k , joka kuuluu joukkoon $S(\mathbf{x}) - T$, ja tuottaa parhaan mahdollisen arvon tavoitefunktiolle $c(\mathbf{x})$. Tabujoukkoa T päivitetään iteraatiokierroksilla siten, ettei päätyminen jo aiempaan ratkaisuun ole mahdollista. Näin ollen algoritmin on mahdollista löytää myös edellistä huonompia ratkaisuja

ja kulkea seuraavaan lokaaliin optimiin. Paras löydetty ratkaisu tallennetaan vektoriin \mathbf{x}^* , jota päivitetään iteraation edetessä tarpeen mukaan. Iteraatio pysäytetään perustuen joko kokonaisiteraatioiden määrään, iteraatioiden määrään \mathbf{x}^* :n päivittämisestä tai jos joukosta $S(\mathbf{x}) - T$ muodostuu tyhjä joukko. (Glover 1989)

Verrattuna edellä esitettyihin optimointimenetelmiin, **geneettiset algoritmit** ovat ennemminkin samantapaisten optimointimenetelmien joukko kuin yksittäinen optimointimenetelmä. Näissä menetelmissä tarkastellaan erilaisia ratkaisujoukkoja, populaatioita. Populaation jokaisella alkiolla, eli muuttujavektorin arvolla, lasketaan tavoitefunktion, jota tässä tapauksessa kutsutaan sopivuusfunktioksi, arvo. Tarkastelupopulaatiosta valitaan välipopulaatio perustuen sopivuusfunktion arvoihin siten, että ratkaisun sopivuus kasvaa todennäköisyyttä tulla valituksi välipopulaatioon. Valintamenettely vaihtelee riippuen algoritmista ja yksittäinen ratkaisu voi tulla valituksi välipopulaatioon useamman kerran. Välipopulaation pohjalta muodostetaan ratkaisuparit ja niiden pohjalta edelleen seuraava populaatio uudelleenyhdistelemällä muodostettavien ratkaisuparien alkioita. Alkiot yhdistellään katkaisemalla muuttujavektori jostakin määrätyllä todennäköisyydellä valitusta kohdasta ja liittämällä vektorien alut ja loput ristiin yhteen. Lopputuloksena saatuja vektoreita voidaan menetelmästä riippuen edelleen modifioida mutaatiotekijällä, joka jollakin todennäköisyydellä muuttaa vektorin muuttujan arvoa. Näin saadaan uusi populaatio, jonka pohjalta prosessi aloitetaan uudelleen. (Whitley 1994) Algoritmin lopputuloksena populaatio kehittyy ja keskittyy kohti optimia. Algoritmi voidaan pysäyttää esimerkiksi populaatioiden määrästä riippuvaisena. Toisaalta voidaan tarkastella populaation sopivuusfunktion tulosten jakaumaa ja pysäyttää algoritmi, kun ratkaisu on riittävän suppea. Olennaista on huomata, että kaikki geneettisiin algoritmeihin perustuvat menetelmät eivät säilytä löytynyttä parasta ratkaisua, vaan antavat tulokseksi viimeisen populaation parhaan ratkaisun. Vaihtoehtoisesti uuteen populaatioon voidaan aina kopioida edellisen populaation paras ratkaisu ilman risteytysprosessia, jolloin paras kohdattu ratkaisu säilytetään lopputuloksena (Jalkanen 2007). Tällöin voidaan algoritmin pysäyttämiseen hyödyntää myös parhaimman ratkaisun löytymistä seurannutta populaatioiden määrää.

Parveilualgoritmissa jokin määrä muuttujavektoreita asetetaan ratkaisuvastuuteen ja ratkaistaan tavoitefunktion arvo. Tavoitefunktion arvojen laskemisen jälkeen jokaisen vektorin alkioiden arvoja muutetaan perustuen vektorin omaan tavoitefunktion arvojen historiaan, muiden vektorien arvoihin sekä ulkoisiin häiriöihin. Edelleen tavoitefunktioiden arvot lasketaan uusilla vektorien alkioiden arvoilla. Iteraatiota toistetaan pysäytysehtoon asti, jolloin on löydetty optimi. Parveilualgoritmissa tarkastellaan partikkeleita, jotka koostuvat kolmesta vektorista: sijaintivektorista \mathbf{x}_i , joka vastaa muuttujavektoria, parhaan tuloksen vektorista \mathbf{p}_i , johon tallennetaan parhaan tavoitefunktion arvon tuottanut sijaintivektori sekä nopeusvektorista \mathbf{v}_i , joka voidaan ajatella siirtymävektoriksi. (Poli et al. 2007)

Rakenteen optimointiin on olemassa kolme erilaista lähestymistapaa: koon optimointi, muodon optimointi sekä topologian optimointi. Koon optimoinnilla tarkoitetaan rakennesosan kokoparametrin, kuten poikkileikkauspinta-alan optimointia siten, että se on mahdollisimman tehokkaasti hyödynnetty. Muodon optimoinnissa rakennesosan muotoa, esimerkiksi rakennekorkeutta muuttamalla pyritään löytämään mahdollisimman tehokas rakenne. Topologian optimoinnissa rakennesosan sisäistä massoittoa, kuten ristikon uumasauvojen sijoittelua ja määrää, muuttamalla pyritään löytämään tehtävän optimiratkaisu. Optimointitehtävä voidaan myös muodostaa edellä mainittujen yhdistelmänä. (Jalkanen 2007)

3.4 Optimointi rakennesuunnittelussa

Rakennesuunnittelun peruselementteinä käytetään tyypillisesti sauvamaisia rakennesosia, kuten palkkeja ja pilareita sekä erilaisia laattoja. Sauvojen osalta hyödynnetään yleensä vakio poikkileikkauksia, teräsrakenteissa valssattuja tai kylmämuokattuja standardiprofiileja ja puurakenteissa materiaalikohtaisia vakiokokoja. Myös betonirakenteissa poikkileikkauksille käytetään tyypillisesti BES:n (Betonielementtistandardi) mukaisia kokoja, vaikkakin paikallavalurakenteissa kokoja voidaan tarpeen mukaan soveltaa. Vakioituista poikkileikkauksista johtuen voidaan todeta, että tyypillinen rakennesuunnittelun optimointitehtävä ei ole jatkuva.

Toinen merkittävä kantavien rakenteiden optimointiin vaikuttava tekijä on käytettävä suunnitteluohjeisto. Suunnitteluohje määrää kriteerit, joiden perusteella rakennesosan kestävyyttä vallitsevia rasiustiloja vastaan tarkastellaan. Näin ollen suunnitteluohje määrää kaavoissa (8) ja (9) esitetyt optimointitehtävän reunaehdot. Suunnitteluohjeet voivat sisältää yksinkertaisia alkeisfunktioiden avulla määriteltyjä ehtoja tai varsin monimutkaisia ehtoja sisältäen epäjatkuvuutta, derivoitumattomuutta sekä osittain diskreettejä määriteltyjä. Nämä tekijät aiheuttavat haasteita optimointitehtävän ratkaisulle.

Yksittäistä, yhdestä sauvasta koostuvaa rakennesosaa tarkasteltaessa voidaan todeta, että optimointitehtävä on tyypillisesti koon ja/tai muodon optimointia riippuen erilaisista mahdollisista rasiustiloista. Koska mahdollisia poikkileikkauksia on tyypillisesti huomattavan rajoittunut määrä, ei näissä tapauksissa yleensä sovelleta optimointimenetelmiä vaan lasketaan vaihtoehtoiset poikkileikkauskoot läpi ja valitaan taloudellisin niistä, jotka kestävät vallitsevat rasitukset. Vasta kun tarkasteluun mukaan otetaan topologian optimointi, esimerkiksi rakennesosien sijoittelun tarkastelussa tai ristikon sauvoittelun tarkastelussa, tulee optimointialgoritmien hyödyntäminen tarkoituksenmukaiseksi, koska muutamankin vaihtoehdon läpilaskeminen on työläs tehtävä.

Kuten edellä todettiin, rakennesuunnittelun optimointitehtävä on tyypillisesti diskreetti tai sekalukutehtävä, jolloin funktion ominaisuuksia hyödyntävien optimointimenetelmien soveltaminen ei tule kyseeseen. Näin ollen optimoinnissa sovelletaan tyypillisesti jotakin

luvussa 3.3 esitettyä optimointimenetelmää tai muuta heuristista menetelmää. Kirjallisuudessa tyypillisesti tarkasteltu vakio-poikkileikkauksia sisältävä optimointiongelma on teräsputkiristikoiden optimointi. Muun muassa Jalkanen (2007), Lamberti (2008) ja Mela (2013) tarkastelevat ristikoiden optimointia erilaisilla heuristisilla optimointimenetelmillä. Tutkimusten perusteella ei voida esittää mitään optimointimenetelmää toista paremmaksi. Optimointimenetelmän soveltuvuus riippuu tehtävän formuloinnista ja valituista muuttujista.

Rakennesuunnittelulle on tyypillistä, että taloudellisimmaksi rakenteeksi oletetaan kevyin kriteerit täyttävä rakenne. Näin ollen tavoitefunktioiksi valitaan tyypillisesti rakenteen massafunktio. Tämän lähestymistavan on valinnut esimerkiksi Jalkanen (2007). Kuitenkin rakenteen kustannuksiin vaikuttaa myös esimerkiksi liitosten toteutus, kuten hitsaus, mahdolliset pintakäsittelyt sekä mahdollinen suojauksen tarve, kuten palonsuojamaalaus. Pavlovčič et al. (2004) esittävät lisäksi, että kustannuksiin vaikuttavat ko-koamiskustannukset (myös mahdolliset ohjurit ja sovittelevyt tai vastaavat), rei'itys ja katkaisukustannukset, kuljetuskustannukset ja rakentamiskustannukset. Pavlovčič et al. (2004) esittävät myös tuloksia, joiden perusteella huomataan, että massaltaan suurempi rakenne saattaa olla kokonaistaloudellisesti kustannustehokkaampi kuin kevyempi ratkaisu. Näin ollen voidaan todeta, ettei massafunktio ole kaikissa tapauksissa pätevä tavoitefunktio.

Toisaalta tavoitefunktiossa on järkevää ottaa huomioon ainoastaan ne tekijät, jotka voivat muuttua eri rakennevaihtoehtojen välillä. Esimerkkinä voidaan tarkastella leikkausliitosta, joka on tarkoitus toteuttaa ruuviliitoksena ja johon vaikuttava leikkausvoima on vakio riippumatta rakenneratkaisusta. Mikäli ruuvien koko riippuu pelkästään lujuste-
kennististä tekijöistä, voidaan ruuveja pitää vakiona riippumatta optimoitavasta rakenneratkaisusta. Tällöin optimointiprosessi ja sen lopputulos ei vaikuta ruuviliitokseen eikä myöskään sen kustannukseen. Näin ollen liitoksen kustannusten sisällyttäminen tavoitefunktioon on turhaa ja kuluttaa ainoastaan laskentatehoa. Haapio (2012) esittää väitöskirjassaan kustannuskeskittymiin perustuvan mallin, jota voidaan soveltaa optimoinnin tavoitefunktion formulointiin. Malli perustuu kustannuskeskittymiin, joiden arvot summataan yhteen kokonaiskustannusten muodostamiseksi. Jokaiselle kustannuksiin vaikuttavalle tekijälle, kuten esimerkiksi katkaisulle ja hitsaukselle on esitetty oma kustannuskeskittymänsä ja kaavat sen laskemiseen. (Haapio 2012) Menetelmä on näin ollen helposti sovellettavissa tapauksissa, jossa osa kustannustekijöistä on vakioita ja ne voidaan näin ollen jättää pois tarkastelusta. Menetelmän soveltamista esitellään lähteessä (Mela 2013).

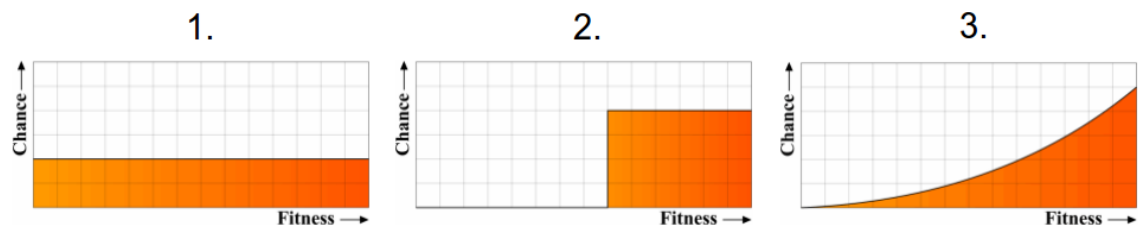
3.5 Grasshopper – Galapagos

Tässä diplomityössä tarkasteltiin optimointia Rhinoceros-ohjelmiston Grasshopper-lisäosan Galapagos-ratkaisijalla. Galapagos on optimointiratkaisija, joka voi ratkaista optimointitehtäviä joko evoluutioalgoritmeilla tai simuloitulla jäähdytyksellä. Ratkaisijoiden

taustalla toimiva optimointialgoritmi on huonosti dokumentoitu, tässä käydään lyhyesti läpi lähdekirjallisuudesta saatavilla olevat tiedot.

Galapagoksen **evoluutioalgoritmi** on geneettinen algoritmi, jonka toiminta alkaa muuttujien ja tavoitefunktion määrittelyllä. Tämän jälkeen luodaan populaatio, jonka koko on käyttäjän määriteltävissä, muuttujien satunnaisilla arvoilla ja lasketaan tavoitefunktion arvot kullekin populaation yksilölle. Tavoitefunktion arvojen laskemisen jälkeen suoritetaan välipopulaation valintamenettely. Galapagos sisältää kolme erilaista perusalgoritmia, sekä itse valinta-algoritmin, joka yhdistelee perusalgoritmeja. Perusalgoritmit ovat:

1. isotrooppinen valinta
2. eksklusiivinen valinta
3. puolueellinen valinta (Rutten 2010).



Kuva 3.2. Galapagos-ratkaisijan valinta-algoritmit (muokattu lähteestä (Rutten 2010)).

Perusalgoritmien toiminta on esitelty kuvassa 3.2. Kuvassa vaaka-akselilla on soveltuvuus (eli miten lähellä yksilön tavoitefunktion arvo on tavoiteltavaa arvoa, esimerkiksi minimiä) ja pystyakselilla soveltuvuus. Isotrooppinen valinta (1.) valitsee välipopulaatioon kaikki yksilöt. Eksklusiivinen valinta (2.) valitsee välipopulaatioon yksilöt, joiden sopivuus on jotakin raja-arvoa parempi. Puolueellinen valinta (3.) kasvattaa välipopulaatioon valituksi tulemisen todennäköisyyttä sopivuuden funktiona siten, että parempi sopivuus tuottaa suuremman todennäköisyyden tulla valituksi välipopulaatioon. (Rutten 2010) Galapagoksen valinta-algoritmi sekoittaa näitä perusalgoritmeja määrittelemättömällä tavalla suorittaessaan välipopulaation valinnan.

Välipopulaation pohjalta suoritetaan risteytys eli ratkaisuparien alkioiden sekoittaminen. Ratkaisuparien valinta suoritetaan perustuen yksilöiden etäisyyteen ratkaisuavaruudessa (jonka ulottuvuus vastaa muuttujavektorin ulottuvuutta). (Rutten 2010) Etäisyydellä tarkoitetaan siis euklidista etäisyyttä n-ulotteisessa avaruudessa, joka voidaan laskea kaavan

$$d(p, d) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} \quad (11)$$

mukaan, missä $d(p, d)$ tarkoittaa pisteiden p ja d välistä etäisyyttä ja p_i ja d_i pisteiden alkioidia. Käyttäjän valitsemalla sisäsiirtoisuustekijällä voidaan säätää miltä etäisyydeltä mahdollinen pari voi löytyä, eli millä välillä tulos $d(p, d)$ voi olla (Rutten 2010). Sitä, millä tavoin pari valitaan sallitusta joukosta, ei ole dokumentoitu.

Kun parit on valittu, ne risteytetään uuden populaation luomiseksi. Galapagoksessa on jälleen useampi perusalgoritmi risteytyksen toteuttamiseen. Ensimmäinen vaihtoehto on luvussa 3.3 esitetty risteytys, jossa uuden sukupolven ominaisuudet muodostetaan siten, että parin alkiovektorit katkaistaan jostakin kohtaa, ja uuden sukupolven yksilöt muodostetaan liittämällä katkaistut osat ristiin. Toinen vaihtoehto on muodostaa uuden sukupolven alkiot keskiarvoilla risteytysparin alkioista. Kolmas vaihtoehto on toteuttaa risteytys kuten toisessa vaihtoehdossa kuitenkin siten, että parin toisen osapuolen arvoja painotetaan keskiarvon laskennassa perustuen sopivuuteen tavoitefunktion näkökulmasta. (Rutten 2010) Kuten edellä, sitä, millaisella algoritmilla risteytysvaihtoehto valitaan, ei ole dokumentoitu.

Kun uusi sukupolvi on muodostettu, sitä voidaan mutatoita ennen tavoitefunktion arvojen laskemista. Ohjelmiston lähdekoodi mahdollistaa mutatoinnin eri keinoin, mutta ohjelman toteutuksessa on tällä hetkellä käytössä vain yksi keino mutatoita yksilöitä. Tässä keinossa mutointi toteutetaan siten, että jotakin yksilön alkion arvoa muutetaan sille sallitulla välillä satunnainen määrä. (Rutten 2010)

Kun edellä esitellyt vaiheet on suoritettu, aloitetaan alusta tavoitefunktion arvon laskennasta kullekin yksilölle. Ohjelmisto mahdollistaa edellisen sukupolven parhaiten sopivien yksilöiden kopioimisen seuraavan sukupolven heikoimpien yksilöiden paikalle, suhteellinen säilytettävien yksilöiden määrä on käyttäjän säädeltävissä. Algoritmi voidaan pysäyttää joko perustuen parhaan ratkaisun pysyvyyteen (ensisijainen vaihtoehto) tai toissijaisesti absoluuttiseen ratkaisijan käynnistämisestä kuluneeseen aikaan.

Galapagoksen **simuloitua jäähdytystä** hyödyntävä algoritmi perustuu simuloitun jäähdytyksen standardimalliin, joka on esitelty lähteessä muun muassa lähteessä (Kirkpatrick et al. 1987). Galapagoksen tarjoamassa algoritmista on mahdollista säätää aloituslämpötilaa, jäähdytysnopeutta sekä seuraavan etsintäpisteen muutosulottuvuutta edelliseen nähden. Aloituslämpötilan säätäminen voidaan toteuttaa välillä 1...100 %, kun 0 % vastaa jäätymisspistettä ja 100 % lämpötilaa, jossa kaikki muutokset systeemin sisällä ovat mahdollisia. Jäähdytysnopeudella kuvataan sitä, miten nopeasti systeemi jäähtyy eli miten nopeasti ratkaisu suppenee. Hitaalla jäähdytyksellä muutokset systeemin sisällä ovat mahdollisia pidempään, mikä parantaa globaalin optimin löytymisen mahdollisuuksia sekä ratkaisun tarkkuutta. Vastaavasti nopealla jäätymisellä löydetään nopeasti lokaaleja optimeja, mutta ratkaisu pysyy todennäköisemmin löytyneessä lokaalissa optimissa. Etsintäpisteen muutosulottuvuus tarkoittaa sitä, että kuinka monta edellisen ratkaisuvektorin alkia voidaan uuteen ratkaisuun muuttaa.

Galapagoksen simuloitun jäähdytyksen ratkaisija aloittaa jäähdytyksen ja ratkaisun löytymisen jälkeen aina uuden ratkaisun ja tallentaa edellisen kierroksen parhaan löydetyn ratkaisun. Tallennettu ratkaisu korvataan uuden kierroksen parhaalla ratkaisulla, mikäli se on tallennettua ratkaisua parempi. Kun ratkaisija pysäytetään, käyttäjälle annetaan tallennettu, eli paras löytynyt ratkaisu.

4. PARAMETRINEN SUUNNITTELU KÄYTÄNNÖN TYÖKALUNA

4.1 Muut suunnittelualat

Kuten luvussa 2 asiaa sivuttiin, parametrinen suunnittelun käsite on riippuvainen sovel-luskohteesta eli siitä, mitä suunnitellaan. Rakennustekniikkaa lähellä olevien ja saman-kaltaisia suunnittelutehtäviä toteuttavien suunnittelutehtäviä voidaan yleistää esimerkiksi taulukonTaulukko 4.1 mukaan.

Taulukko 4.1. Suunnittelualojen pääasialliset tehtävät.

Suunnitteluala	Suunnittelutehtävä
ARK	Rakennuksen tilat ja muoto
RAK	Rakenneosan kapasiteetti (ja muoto)
LVI	Putkiston muoto ja kapasiteetti
Kone	Koneen/laitteen/osan muoto ja kapasiteetti

TaulukossaTaulukko 4.1 on esitetty karkeasti kunkin suunnittelualan pääasialliset tehtä-vät. Vaikka kyseessä on karkea yleistys, päästään tällä kiinni siihen, mitä parametrisella suunnittelulla voidaan tarkoittaa.

Rakennussuunnitteluun liittyvistä suunnittelualoista parametrsta suunnittelua hyödyntä-vät tällä hetkellä eniten arkkitehdit. Arkkitehdin ensisijainen tehtävä on suunnitella ra-kennuksen muoto. Näin ollen parametrsta suunnittelua käytetään tyypillisesti rakennuk-sen muodon suunnitteluun. Tyypillisiä suunnittelumenetelmiä ovat muodon generoimi-nen kokonaan parametrisilla algoritmeilla ja alustavasti määrättyjen muotojen linkittämi-nen toisiinsa parametrisilla reunaehdoilla (Woodbury 2010). Nämä antavat suunnitteli-jalle mahdollisuuden tutkia erilaisia muotoja, massojen tai tilojen sijoittelua ja suunnitte-luratkaisujen toimivuutta (Woodbury 2010).

Maailmalta löydetään useita merkittäviä rakennuskohteita, joiden arkkitehtisuunnitte-lussa on hyödynnetty parametrsta suunnittelua. Vaikuttava esimerkki on Kiinan Maca-oon rakennettava The Morpheus Hotel, jonka suunnittelu on toteutettu suurimmilta osin Rhinoceros/Grasshopper -ohjelmistoilla. Rakennus toteutetaan kantavana runkorakentee-naan rakennuksen pintaverhouksen ulkopuolella oleva teräsrakenteinen tukiranka, joka näkyy myös kuvasta 4.1, sekä rakennuksen sisäpuolinen liittorakenteinen runko.



Kuva 4.1. *The Morpheus Hotel (Zaha Hadid Architects)*

Morpheus-hotellin suunnitteluprosessi eteni vaiheittain parametrisena prosessina. Ensin rakennuksen ulkomuodon geometria määriteltiin parametrisesti pohjana suorakulmainen särmiö, jonka geometria voidaan nähdä kuvasta 4.1. Keskelle muodostettiin parametrisesti aukot siten, että kannasten kohdalla rakennuksen rungon kaksi pystykuilua yhdistyvät yhtenäisillä välipohjatasoilla. Määritellyn ulkomuodon päälle luotiin tämän jälkeen parametrisella työkalulla verkko. Verkon mukaan rakenteelle optimoitiin teräsrakenteinen putkirunko. Putkirungon päälle suunniteltiin edelleen alumiinikuori, jolla teräsputket verhoiltiin. Kaikki edellä mainitut suunnitteluvaiheet toteutettiin parametrisesti siten, että parametrinen mallin lopputuloksena saatiin kunkin kappaleen geometrinen malli liitoksia myöten. (McNeel North America 2017)

Myös koneensuunnittelussa hyödynnetään parametrissa suunnittelua. Suunnittelualalle on tyypillistä, että suunniteltavan osan tai rakenteen toiminnallisuus on etusijalla ja muoto suunnitellaan palvelemaan toiminnallisuutta. Tyypillisesti tämä tarkoittaa sitä, että suunnittelija tekee alkuarvauksen suunniteltavan osan muodosta perustuen sen toiminnallisuuteen. Alkuarvauksesta voidaan nyt parametrisoida toiminnallisuuteen vaikuttavat tekijät, kuten reiän halkaisija tai levyn paksuus. (Myung & Han 2001) Parametreja varioimalla voidaan edelleen etsiä paras ratkaisu perustuen valittuihin suunnittelukriteereihin esimerkiksi joko toiminnallisuutta arvioimalla tai hyödyntämällä optimointia esimerkiksi massa suhteen.

Arkkitehtisuunnittelun ja koneensuunnittelun välillä on merkittävä ero. Vaikka molempien suunnittelualojen yhteydessä puhutaan parametrisesta suunnittelusta, ovat tavoitteet ja näin ollen myös menetelmät täysin erilaisia. Arkkitehti pyrkii löytämään muodoltaan sopivan ratkaisun visuaalisten kriteerien, tilaohjelman, kustannusten sekä käytettävyyden perustella, kun taas koneensuunnittelija etsii parhaiten toimivaa ratkaisua kriteerinään esimerkiksi rakenteelliset ominaisuudet kuten lujuus ja taloudelliset ominaisuudet kuten massa. Merkittävänä erona on se, että miellyttävä visuaalinen ulkomuoto on yksi olennainen arkkitehtuurin tavoite.

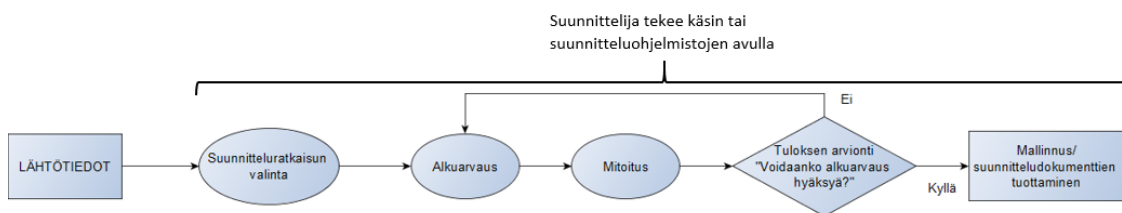
Muilla suunnittelualoilla parametrisen suunnittelun hyödyntäminen on kirjallisuuden perusteella olematonta. Alalla on tehty jonkin verran tutkimustyötä, muun muassa (Pfafferoth et al. 2003), mutta parametrisoinnin hyödyntämistä perussuunnittelussa ei ole raportoitu. Kuitenkin seuraavan luvun perusteella voidaan pitää itsestään selvänä, että esimerkiksi LVI-suunnittelussa tehdään jossain määrin myös parametrissa suunnittelua, vaikkei sitä välttämättä kutsuta tällä nimellä. Tämä perustuu siihen, että joka tapauksessa suunnittelu edellyttää esimerkiksi vaihtoehtoisten putkistoreittien tutkimista, jolloin parametreina voi olla esimerkiksi eri huoneiden tarvitsema ilmamäärä. Toisaalta voidaan nähdä suuri potentiaali parametrisen suunnittelun tietoiselle kehittämiselle juuri tällä alalla.

4.2 Parametrinen suunnittelu käytännössä

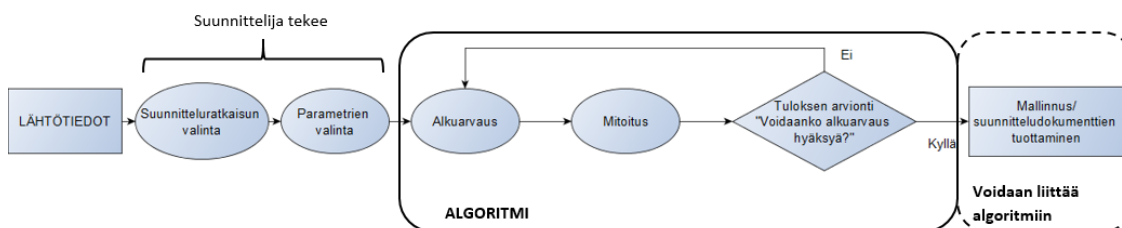
Kuten edellä on esitetty, suunnittelutehtävän parametreiksi on mahdollista valita geometrisen reunaehdotehtävän reunaehdoja. Rakennesuunnittelun tapauksessa tyypillisesti esimerkiksi arkkitehti on jo ennalta määrännyt osan geometrisista reunaehdoista.

Määriteltäessä parametrissa kantavien rakenteiden suunnittelua huomataan, että se on periaatteeltaan täysin sama asia kuin perinteisellä tavalla toteutettu kantavien rakenteiden suunnittelu. Käytettäessä termiä parametrinen suunnittelu tarkoitetaan, että osa suunnittelijan tehtävistä toteutetaan tai ratkaistaan tietokoneavusteisesti. Kuvassa 4.2 on havainnollistettu perinteisen rakennesuunnittelun ja parametrisen suunnittelun välisiä eroja.

Perinteinen suunnittelumenetelmä



Parametrinen suunnittelumenetelmä



Kuva 4.2. Perinteisen suunnittelumenetelmän ja parametrisen suunnittelun välinen ero.

Kuvassa on esitetty sekä perinteinen että parametrinen suunnittelumenetelmä prosessi-kaavioina. Suunnittelu alkaa lähtötietojen pohjalta, jotka saadaan suunnittelutehtävästä riippuen yleensä joko arkkitehdiltä, tilaajalta tai molemmilta. Lähtötietojen valitaan suunnitteluratkaisu eli kantavien rakenteiden suunnittelussa kuormankantotapa. Nämä vaiheet ovat tyypillisesti molemmissa menetelmissä samat. Toki on mahdollista toteuttaa algoritmi, joka valitsee vaakarakenteen esimerkiksi palkin ja ristikon välillä jännevälin funktiona, mutta tässäkin tilanteessa on valittu, että kuorma kannetaan nimenomaan vaakarakenteella.

Kun suunnitteluratkaisu on valittu, valitaan parametriseissa suunnitteluprosessissa parametrit, joiden perusteella suunnittelu tehdään. Parametrien valinta on monitahoinen tehtävä, joka riippuu halutusta lopputuloksesta. Sitä käsitellään tarkemmin luvussa 5.3. Se kuitenkin perustuu vastaaviin asioihin, joiden pohjalta suunnittelija tekee alkuarvauksen perinteisessä suunnittelumenetelmässä.

Seuraavana vaiheena molemmissa prosesseissa on alkuarvauksen tekeminen. Varsinkin yksinkertaisissa rakenteissa, kuten palkeissa ja pilareissa kokenut suunnittelija osaa yleensä valita rakennusosan koon suuruusluokkana oikein, jota voidaan käyttää alkuarvauksena. Tästä edetään mitoitukseen, jossa tutkitaan kaikki rakenneosalle määrätyt suunnittelukriteerit kuten kapasiteetti murtorajatilassa ja siirtymät käyttörajatilassa. Tästä edetään rakennusosan kelpoisuuden varmistamiseen eli tarkastellaan täyttääkö alkuarvaus kaikki sille asetut kriteerit. Mikäli kriteerit eivät täyty, korjataan alkuarvausta mitoituksessa saatujen tulosten perusteella ja käydään uudelleen läpi samat vaiheet, kunnes saavutetaan hyväksyttävä lopputulos. Kun hyväksyttävä lopputulos on saavutettu, suunnittelija piirtää tai mallintaa lopputuloksen, jotta se voidaan toteuttaa työmaalla.

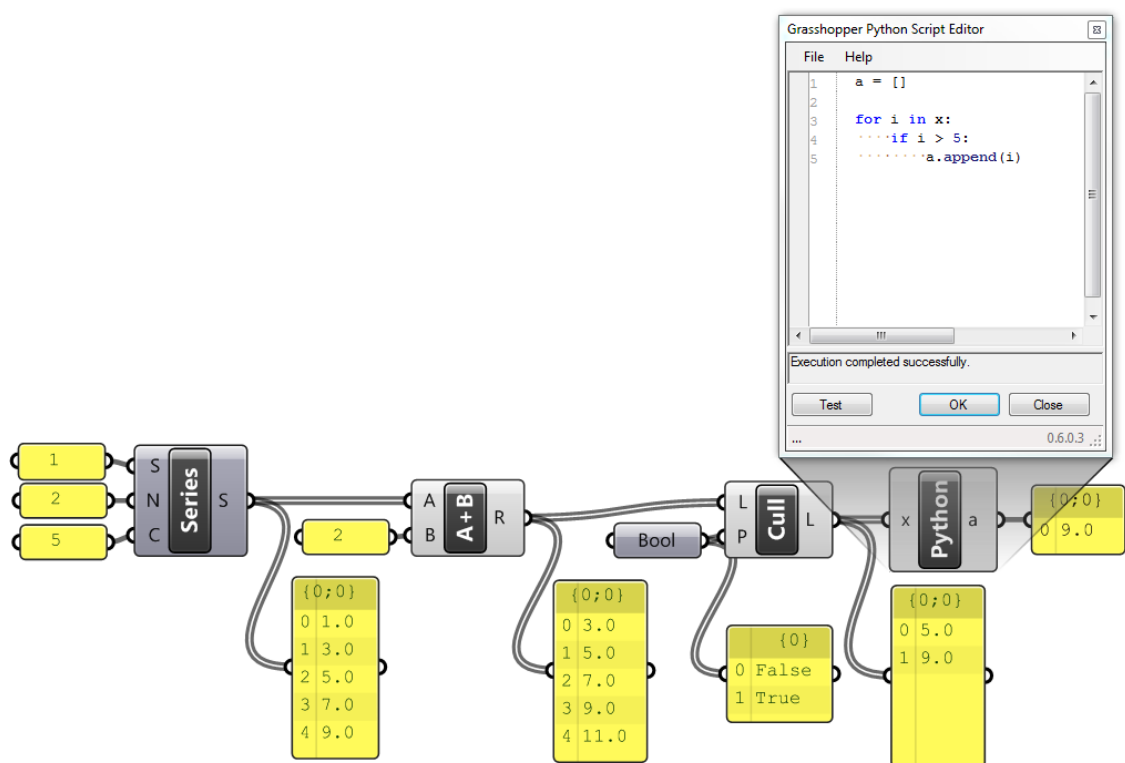
Parametrisessa suunnittelussa alkuarvaus generoidaan algoritmin avulla heti hyväksyttäväksi. Algoritmi käy läpi samat vaiheet kuin suunnittelija, mutta tekee sen automaattisesti. Olennaista on kuitenkin huomata, että parametrien valinta on suoritettava siten, että hyväksyttävä lopputulos on saavutettavissa. Esimerkiksi jos mitoituskriteerinä on palkin taivutuskapasiteetti, tulee parametrien avulla pystyä vaikuttamaan tähän jotakin palkin ominaisuutta muuttamalla, jotta ratkaisu olisi löydettävissä.

Kun käytettävä suunnitteluratkaisu on valittu, tulee se vielä dokumentoida tuotantoa varten. Nykyään se tarkoittaa käytännössä rakenneosan mallintamista valitulla rakennemallinnusohjelmalla. Perinteisessä suunnittelumenetelmässä tämä tehdään yleensä käsin. Algoritmiavusteisissa suunnitteluohjelmistoissa on kuitenkin mahdollista linkittää suunnitteluohjelmiston algoritmi suoraan mallinnusohjelmistoon ja näin hyödyntää jo olemassa olevaa tietoa suoraan ilman ylimääräisiä työvaiheita. Lisäksi tämä mahdollistaa tietomallinnuksessa tarvittavan rakenneosan kohdistuvan lisätiedon parametrisoinnin. Tällaista voi olla esimerkiksi pintakäsittelyn riippuvuus rakennuksen kerroksesta.

Perinteisen suunnittelumenetelmän ja parametrisen suunnittelumenetelmän erona on siis parametrien valinnan lisäksi käytettävä algoritmi. Algoritmin määritelmän perusteella myös perinteisessä suunnittelumenetelmässä on algoritmi, mutta tässä tarkoitetaan algoritmilla nimillä nimenomaan ohjelmistopohjaista algoritmia.

Edelleen voidaan tämän määritelmän perusteella todeta, että on olemassa paljon erilaisia parametrisen suunnittelun ohjelmistoja, joita ei välttämättä mielletä parametrisiksi, koska algoritmia ei ole luotu itse. Pääperiaatteina tällaisissa ohjelmistoissa on kuitenkin, että suunnittelija on valinnut jonkin suunnitteluratkaisun ja tietyn määrän kiinteitä reunaeh-toja (esimerkiksi tukiehdot) sekä muuttuvia reunaeh-toja eli parametreja (esimerkiksi poikkileikkaus ja jänneväli), joiden perusteella etsitään sopivaa ratkaisua. Tyypillisesti soveltuvan ratkaisun tärkeimpänä kriteerinä heti rakenteellisen kestävyuden ja toimivuuden jälkeen on taloudellisuus, joka monesti tarkoittaa mahdollisimman pientä materiaali-määrää.

Parametrinen suunnittelu käsitetään yleensä siten, että suunnittelija luo itse algoritmin jollakin osaamallaan menetelmällä. Koska ohjelmointitaito on yleensä rakennesuunnitelijoilla puutteellista, voidaan käyttää visuaalisen ohjelmoinnin (Rutten 2012) työkaluja, joissa ohjelmointikieli on korvattu visuaalisilla komponenteilla. Näihin komponentteihin syötetään lähtötiedot ja saadaan komponentista riippuen jonkinlainen syöte. Tällaisia työkaluja ovat esimerkiksi Rhinoceros plug-in Grasshopper sekä Autodesk-tuoteperheeseen kuuluva Dynamo (Akos et al. 2015; Autodesk).



Kuva 4.3. Esimerkki visuaalisesta ohjelmoinnista Grasshopper-ympäristössä.

Kuvassa 4.3 on esitetty esimerkkinä lyhyt algoritmi Grasshopper-ympäristössä. Kukin kuvassa esitetty komponentti on visuaalisen ohjelman osa, joka suorittaa sille ohjelmoidun tehtävän. Ensimmäinen komponentti ”Series” muodostaa lukujonon, jonka syötteellä ”S” kuvataan jonon ensimmäistä arvoa, ”N” kertoo lukujen välimatkan ja ”C” lukujen määrän, jolloin saadaan tuloksena kuvassa esitetty lukujono. Komponentti ”A+B” suorittaa yhteenlaskun lisäämällä lukujonon ”A” arvoihin luvun ”B”. ”Cull”-komponentti valitsee listan ”L” luvuista kaavan ”P” mukaiset arvot. Viimeinen komponentti ”Python” kuvaa käyttäjän Python-ohjelmointikielellä luomaa komponenttia, joka valitsee annetusta lukujonosta kaikki lukua 5 suuremmat luvut. Kaikki edellä esitetyt komponentit siis sisältävät niille ennalta ohjelmoidun tehtävän esitettyä ”Python”-komponenttia vastaavasti, joskin muulla ohjelmointikielellä toteutettuna. Tällaisten ympäristöjen käyttö on ohjelmointia nopeampaa oppia, ja ne mahdollistavat ohjelmointitaidottomalle suunnittelijalle helposti lähestyttävän tavan luoda algoritmeja.

Algoritmeja ja edelleen johonkin käyttötarkoitukseen sopivia parametrisia suunnittelutyökaluja voidaan kuitenkin luoda myös perinteisillä ohjelmointikielillä. Käytännössä tämä mahdollistaa parametristen suunnittelumenetelmien soveltamisen mihin tahansa rakenteisiin tai rakennesiin. Reunaehtoina toteutukselle on ohjelmoijan ja ohjelmointikielen rajoitteet sekä ohjelmiston tuottamisen taloudellinen kannattavuus. Ohjelmointiin liittyvät asiat vaativat tietoteknistä asiantuntemusta eikä niihin tässä työssä näin ollen perehdytä.

4.3 Parametrisen suunnittelun taloudellisuus ja tehokkuus

4.3.1 Taloudellisuus

Parametrisen mallin luominen suunniteltavasta rakenteesta lisää suunnittelijan työtä lyhyellä aikajänteellä, sillä sopivien parametrien etsiminen ja algoritmien muodostaminen vievät aikaa, joka ei välttämättä näy heti nopeampana suunnitteluna. Näin ollen ensimmäinen suunnitelma tai suunnitelmaluonnos saattaa olla hitaampi toteuttaa, kun otetaan huomioon parametrisen mallin luomiseen kulunut aika. Rakennesuunnittelijan ajankäytön osalta parametrisen suunnittelutyön hyöty alkaa näkyä vasta suunnittelun edetessä.

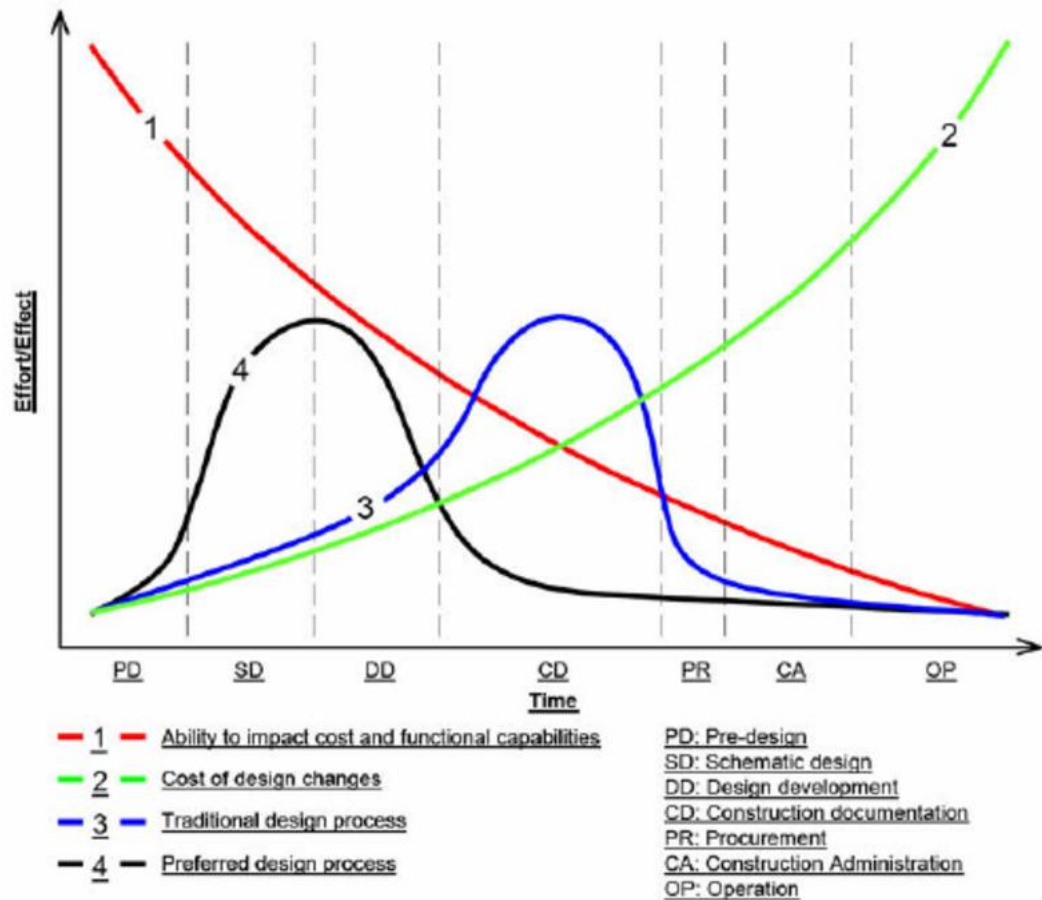
Yksinkertaisimmillaan parametrisen menetelmän hyöty on **geometrialtaan monimutkaisissa rakenteissa**, jotka kuitenkin ovat matemaattisesti määrättyjä. Mikäli arkkitehdin suunnittelema rakenne on geometrisilta muodoiltaan monimutkainen, mahdollistaa parametrinen mallintaminen tällaisten rakenneosien tehokkaan mallintamisen matemaattisten riippuvuuksien avulla. Esimerkkinä voidaan ajatella esimerkiksi kaksoiskaarevaa kattopintaa, joka on tarkoitus kannattaa kattoristikkojen avulla. Tällaisessa jokainen kattoristikko on erilainen ja jokaisen geometrian ja topologian suunnittelu on työläs tehtävä erityisesti suurissa määrissä. Algoritmien avulla taas voidaan määrittää tasot, joita paarteet seuraavat, jako sekä säännöt uumasauvojen muodostusta varten. Tällöin työmäärä on sama riippumatta kattopinnan muodosta tai ristikkojen määrästä.

Tehokkuus haastavasta geometriasta muodostuvan rakenteen suunnittelussa kasvaa, mikäli suunnittelussa voidaan tehdä yhteistyötä arkkitehdin kanssa. Tämä tarkoittaa sitä, että arkkitehti toimittaa rakenteen muodon määräävän geometrian, edellisen esimerkin tapauksessa kattopinnat, sellaisessa muodossa, että rakennesuunnittelija pystyy hyödyntämään niitä omassa ohjelmistossaan. Tällöin rakennesuunnittelijan ei tarvitse mallintaa uudelleen arkkitehdin suunnitelmia, vaan ainoastaan täydentää niitä rakennesuunnitelmilla. Arkkitehdin ja rakennesuunnittelijan välistä yhteistyötä on käsitelty tarkemmin luvussa 5.5.

Toinen merkittävä tilanne, jolloin parametrisestä suunnittelusta voidaan saavuttaa taloudellista hyötyä, on kun suunnitelmiin on **odotettavissa paljon muutoksia**. Tällaisia tilanteita ovat esimerkiksi suunnittelukilpailut, joissa eri suunnittelijoiden yhteistyöllä luodaan suunnitteluehdotus järjestettyyn kilpailuun. Yhteistyön aikana suunnitelmat muuttuvat ja kehittyvät kohti kollektiivisen näkemyksen mukaista parasta ratkaisua. Tämä tarkoittaa lukuisia muutoksia suunnitelmiin uusien asioiden ilmentyessä. Myös tavallisiin suunnittelukohteisiin saattaa tulla suunnittelun aikana paljon muutoksia. Näiden ennustaminen on kuitenkin hankalaa.

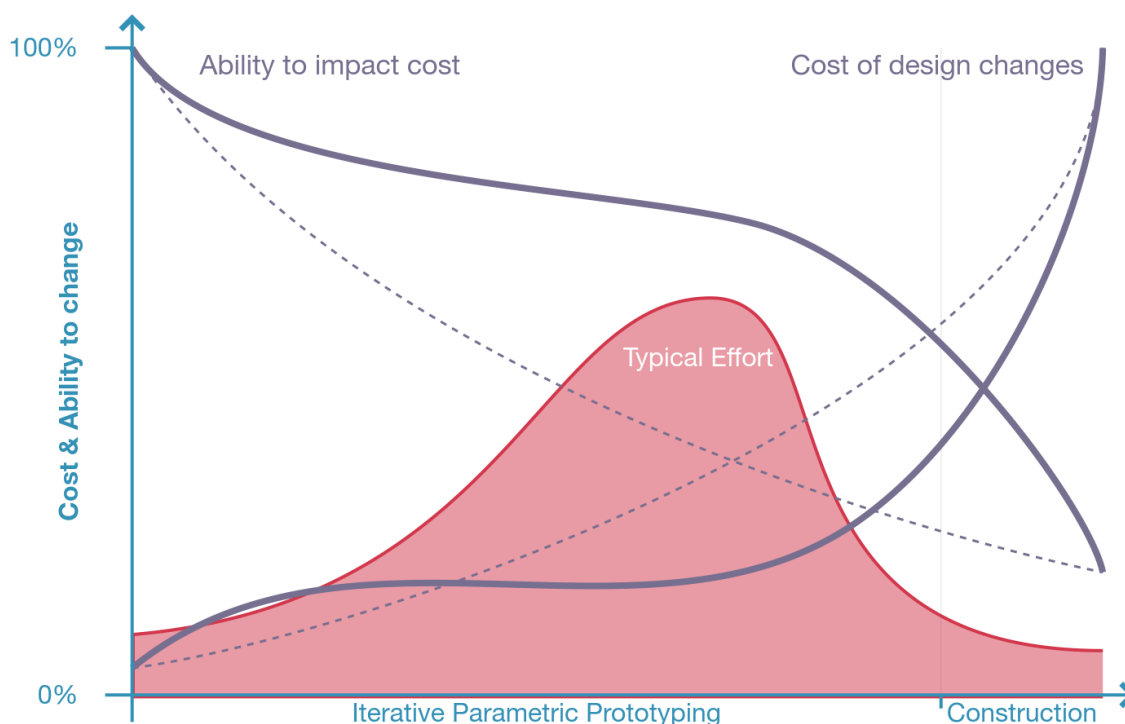
Parametrisen suunnittelun hyöty korostuu muutosten toteutuksessa (Tanska & Österlund 2014). Kun parametrit on valittu sopivasti, suunnitelmia voidaan muuttaa tarvittavilta osin parametreja muuttamalla, jolloin varsinaisen suunnittelutyön osuus jää vähäiseksi.

Myös Davis (2013) on tutkinut aihetta ja päätenyt samaan lopputulokseen. Tyypillisesti muutosten ja kustannusten suhde projektin etenemiseen on esitetty MacLeamyn kuvaajalla (kuva 4.4).



Kuva 4.4. MacLeamyn kuvaaja (CURT 2004).

Kuvasta 4.4 nähdään mahdollisuudet vaikuttaa projektin kustannuksiin ja toiminnollisuuksiin sekä muutosten aiheuttamat kustannukset projektin etenemisen funktiona luonnossuunnittelusta käyttöönottoon. Lisäksi on esitetty perinteisen sekä tavoiteltavan suunnitteluprosessin kulku. Kuvasta havaitaan, että muutosten aiheuttamat kustannukset kasvavat suunnitteluprosessin edetessä, kun taas vaikuttamismahdollisuudet pienenevät. Tämä johtuu siitä, että eri osapuolet edistävät jatkuvasti projektia omilla tahoillaan, jolloin mitä myöhemmässä vaiheessa suunnitelmamuutos tehdään, sitä enemmän se vaikuttaa kyseenomaisen suunnittelualan lisäksi myös muiden osapuolten työhön. Davis esittää (2013), että mikäli koko projektin suunnittelu toteutetaan parametrisesti, voidaan edellä esitetty käyrä muuttaa kuvassa 4.5 esitettyyn muotoon.



Kuva 4.5. Davisin (2013) parametrille suunnitteluprosessille modifioima MacLeamyn käyrä.

Kuvassa 4.5 esitetään, että parametrisen suunnitteluprosessin avulla mahdollisuudet vaikuttaa projektin kustannuksiin tehokkaasti ovat olemassa pidempään verrattuna kuvaan 4.4. Vastaavasti muutosten aiheuttamat kustannukset ovat pienempiä pidemmän aikaa verrattuna kuvaan 4.4. Päätelmää voidaan pitää oikeana, kun tarkastellaan yksittäistä suunnittelualaa ja oletetaan, että merkittäviä (parametrin mallin rikkovia) muutoksia suunnitelmiin ei tehdä (ks. luku 4.3.2). Kuten edellisessä kappaleessa on esitetty, rakennuksen suunnitteluprosessi on monimutkainen kokonaisuus, johon liittyy useita suunnittelualoja sekä muita toimijoita. Näin ollen parametrin suunnittelun soveltaminen jollakin suunnittelualalla ei riitä MacLeamyn käyrän muuttamiseksi. Voidaan jopa väittää, että ilman toimivaa suunnittelunohjausta parametrin suunnittelun soveltaminen jollakin suunnittelualalla lisää koko projektin kustannuksia, jos muutosten tekeminen myöhäisessä koetaan liian helpoksi, koska tällöin aiheutuvat kustannukset kasaantuvat muille toimijoille. Voidaan kuitenkin todeta, että parametrin suunnittelun soveltaminen alentaa yksittäisen suunnittelualan kustannuksia kuvan 4.5 mukaisesti.

Kolmas tilanne, jossa parametrisesta suunnittelusta on mahdollisesti taloudellista hyötyä, on **kahden tai useamman suunnitteluratkaisun vertailu keskenään**. Perinteisellä menetelmällä suunnitteluratkaisujen välinen vertailu vaatii, että kukin suunnitteluratkaisu suunnitellaan vähintään luonnossuunnitelmien tasolle. Tällöin päästään käsitykseen massoista ja määräistä sekä suunnitelman muista ominaisuuksista, joiden perusteella tilaaja voi tehdä päätöksen.

Parametrisoimalla vaihtoehtoisten suunnitteluratkaisujen erot, voidaan suunnitteluratkaisujen vertailu toteuttaa yhdellä parametrisella mallilla. Tämä pienentää työmäärää erityisesti, jos vertailtavia suunnitteluratkaisuja on useita. Lisäksi se mahdollistaa tarkasteltavaksi valittujen suunnitteluratkaisujen välissä olevien ratkaisujen tutkimisen ja optimoinnin esimerkiksi runkomateriaalin määrän suhteen.

Neljäs tapaus, jossa parametrisesta suunnittelusta voidaan saavuttaa hyötyä, on tilanne, jossa **parametrinen malli on helppo ja nopea luoda**. Tällä tarkoitetaan tilannetta, jossa tarvittavat mallin osat tai algoritmit ovat olemassa ja suoraan käytettävissä tai nopeasti muokattavissa suunnittelukohteeseen sopivaksi. Tällöin suunnittelumenetelmästä saavutettava hyöty ei ole yksiselitteisesti todettavissa ennen suunnittelua, elleivät olemassa oleva materiaali tarjoa niin hyvää pohjaa, että suunnittelu on yksinkertaisesti nopeaa vanhaa parametrissa mallia hyödyntämällä, kuin uuden suunnittelu perinteisellä menetelmällä. Muussa tapauksessa suunnittelun tehokkuus perustuu siihen, että aiemmissa tapauksissa esitettyjä hyötyjä voi realisoida odottamatta. Kuten aiemmin esitettiin, suunnittelun ja rakentamisen edetessä saattaa syntyä yllättäviä tarpeita muuttaa suunnitelmia. Mikäli suunnittelu on toteutettu parametrisesti, on näiden muutosten toteuttaminen tehokasta, vaikkei niitä suunnittelutyötä aloitettaessa ole osattu ennakoida.

4.3.2 Tehokkuus

Tanska & Österlund (2014) esittävät, että algoritmiaivusteinen suunnittelu on perinteisiä suunnitteluprosesseja nopeampaa, mutta ajankäytön jaksottuminen projektin eri vaiheille on selvästi erilaista. Alla esitetään perusteluja tämän näkemyksen puolesta ja vastaan sekä eritellään, miten tämä tehokkuus on saavutettavissa.

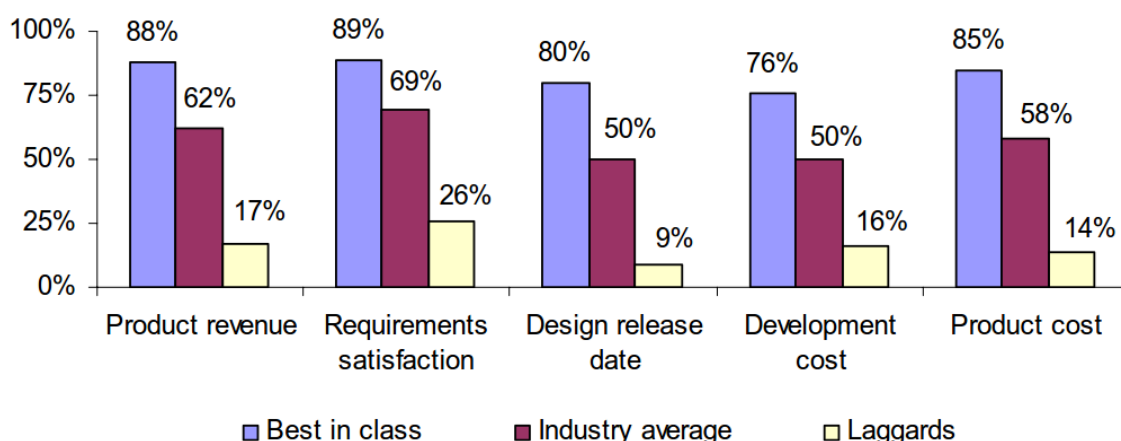
Parametrisen mallin muodostamiseen kuluvaa aikaa on mahdollista lyhentää luomalla ja uudelleenhyödyntämällä algoritmeja, joilla voidaan toteuttaa suunnitteluun liittyviä tyyppisiä tehtäviä (Woodbury 2010). Tällaisten algoritmien, niiden osien tai valmiiksi luotujen komponenttien uudelleenhyödyntäminen mahdollistaa tehokkaan parametrisen mallintamisen. Tämä johtuu yksinkertaisesti siitä, että saman lopputuloksen aikaansaamiseksi tarvitaan vähemmän työtä (Lim 1994). On myös havaittu, että tietokoneavusteisten suunnitteluohjelmien (CAD) käyttäjille on luontevaa pyrkiä löytämään olemassa olevaa aineistoa, jota he voivat hyödyntää työssään (Gantt & Nardi 1992). Tällaista aineistoa pyritään hyödyntämään esimerkiksi (soveltaen lähteestä (Gantt & Nardi 1992)):

- muokkaamalla olemassa olevaa materiaalia
- hyödyntämällä suoraan olemassa olevaa materiaalia
- hankkimalla apua muilta käyttäjiltä.

Edellä mainituista huomataan, että suunnittelijan luonnolliset pyrkimykset puoltavat uudelleenhyödynnettävän materiaalin luomista. Toisaalta voidaan esittää vastaväite, että juuri näiden luontaisten pyrkimysten vuoksi pyritään luomaan tällaista materiaalia,

vaikkei siitä välttämättä ole hyötyä. Myös Woodbury (2010) esittää, että parametrissa mallintamista tekevät suunnittelijat ovat valmiita käyttämään huomattavan määrän aikaa olemassa olevan materiaalin löytämiseen. Woodbury (2010) jättää kuitenkin mainitsematta, kuluuko etsimiseen enemmän aikaa kuin vastaavan materiaalin luomiseen alusta alkaen. Näin ollen materiaalin, niiden algoritmien, komponenttien, kuin vastaavienkin uudelleenhyödyntämiselle tulisi osoittaa perusteet, eikä tyytyä olettamukseen, että uudelleenhyödyntämisen on lähtökohtaisesti oltava tehokasta.

Aberdeenin (2007) toteuttama markkinatutkimus osoittaa kuvan 4.6 mukaisesti, että pyrkimys mallien uudelleenhyödyntämiseen tuottaa organisaatioille taloudellista hyötyä.



Kuva 4.6. Keskeiset suorituskvyn mittarit eri tavoin suunnitelmiaan uudelleenhyödyntävissä organisaatioissa (Aberdeen 2007).

Kuvassa 4.6 on esitetty organisaatioiden keskeisiä suorituskvyn mittareita jaoteltuna parhaiten mallejaan uudelleenhyödyntäviin organisaatioihin (sininen, 20 % organisaatioista), keskitasoon (punainen, 50 %) ja heikoimpiin (keltainen, 30 %) (yhteensä 100 %). Kuvasta havaitaan, että pyrkimys suunnitelmien ja mallien uudelleenhyödyntämiseen korreloi vahvasti organisaation suorituskvyn kanssa. (Huomattavaa on, että kuvassa esitetään tavoitteiden saavuttamisprosentti, joka ei sinällään kuvaa asetettujen tavoitteiden tasoa, jotka voivat vaihdella organisaatioiden välisesti.) Myös aikaisemmin toteutettu seurantatutkimus (Lim 1994) tukee näitä tuloksia. On huomattava, että nämä organisaatiot myös panostavat siihen, että uudelleenhyödyntäminen olisi tehokasta (Aberdeen 2007).

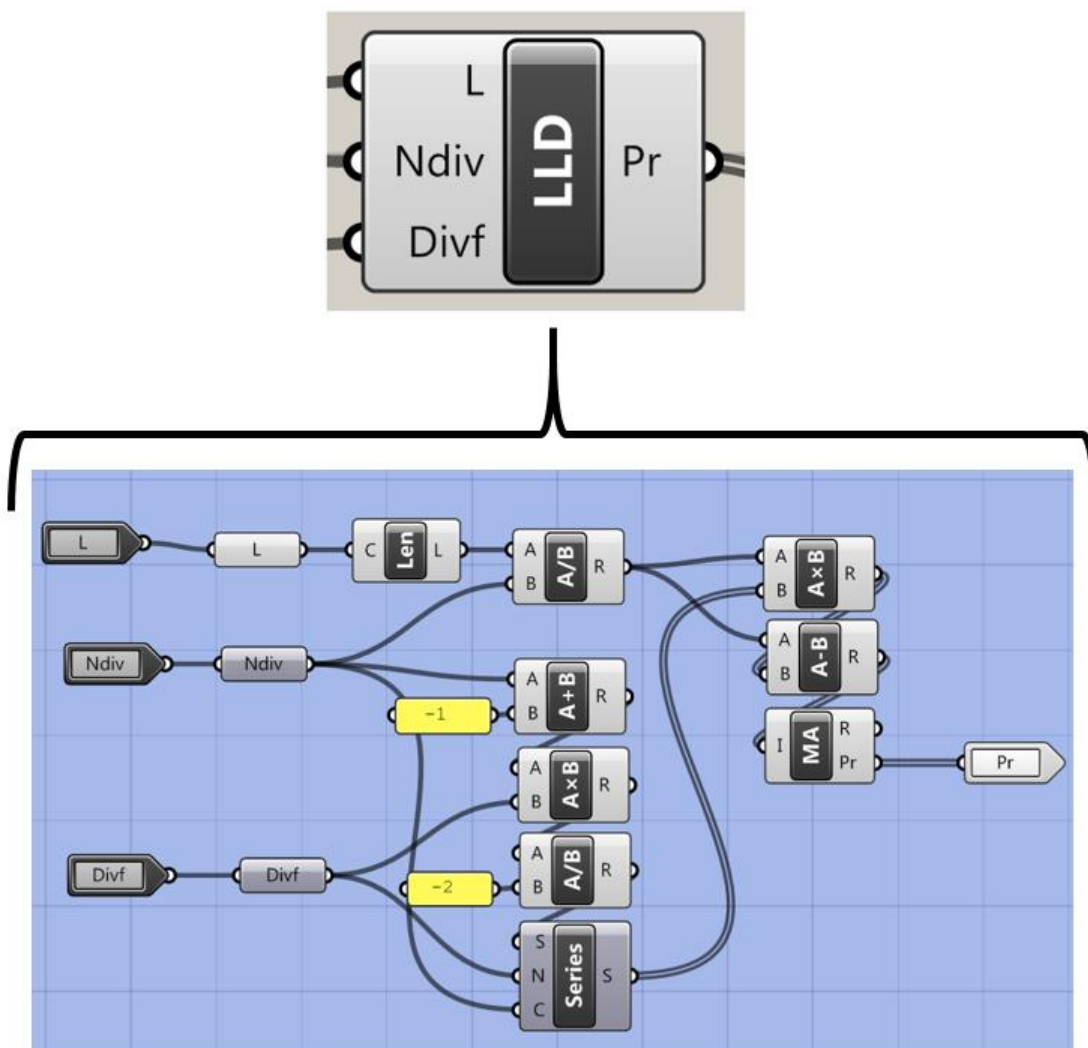
Toisaalta uudelleenhyödynnettävän materiaalin luominen on todettu haastavaksi (Smith 2007; Aberdeen 2007). Smithin (2007) mukaan erityisesti kokonaisen, toimivan algoritmin uudelleenhyödyntäminen on hankalaa, koska muiden kuin algoritmin alkuperäisen luoja on vaikea ymmärtää logiikkaa algoritmin takana. Lisäksi suunnittelijat ovat haluttomia luomaan sellaisia algoritmeja joiden uudelleenhyödyntäminen olisi helppoa (Woodbury 2010).

Aberdeenin (2007) suunnitelmien ja mallien uudelleenhyödyntämistä käsittelevän markkinatutkimuksen mukaan suurimpia hyötyjä uudelleenhyödyntämisestä saaville organisaatioille on yhteistä:

- resurssien kohdentaminen luotujen mallien uudelleenhyödyntämisen valmisteluun
- uudelleenhyödyntämis-näkökulmien tarkastelu suunnitteluvaiheessa
- tehokkaat menetelmät uudelleenhyödynnettävien mallien etsintään
- uudelleenhyödyntämiseen liittyvän ajankäytön seuranta.

Alla on esitetty, miten näitä näkökulmia voidaan soveltaa parametrisessa mallintamisessa. Käytännössä kaikkien toimintatapojen toteuttaminen vaatii resurssien kohdentamista uudelleenhyödyntämisen mahdollistamiseksi. Lim (1994) esittääkin yleisellä tasolla, että uudelleenhyödynnettävän koodin luomisen kustannukset ovat 110...480 % verrattuna tavalliseen, mutta uudelleenhyödyntämisen kustannukset ovat 10...65 % verrattuna täysin uuden koodin luomiseen. Mikäli uudelleenhyödyntäminen toteutetaan alla esitettyjen komponenttien avulla, on niiden luomisen kustannus Lenz et al. (1987) mukaan enintään 100 % suurempi kuin pelkän koodin luomisen.

Paras tapa mahdollistaa koodin uudelleenhyödyntäminen on valmiiden, yhtä tehtävää suorittavien, komponenttien luominen (Woodbury 2010), eli toisin sanoen algoritmin osien luominen lähtökohtaisesti siten, että ne ovat uudelleenhyödynnettävissä. Lenz et al. (1987) vertaa näitä komponentteja rakennusosiin: koodi rakennetaan komponenteista, kuten rakennus rakennetaan rakennusosista. Tässä diplomityössä on valittu käytettäväksi termiä komponentti, joka on helposti ymmärrettävä käytettäessä visuaalisia ohjelmointikieliä. Yleisesti ohjelmointikielissä termien käyttö vaihtelee ja niillä on nyanssieroja, tyyppillisesti esimerkiksi funktio on lähes vastaava termi, mitä tässä tarkoitetaan komponentilla. Vastaavasti Grasshopper-ympäristössä käytetään termiä ”Cluster”, klusteri. Komponentti siis koostuu valitun ohjelmointikielen osista. Sille annetaan lähtöarvot (input), se suorittaa sille ohjelmoidun tehtävän ja tuottaa lopputuloksen (output). Voidaan siis sanoa, että komponentti sisältää kapseloidun toiminnallisuuden (Lenz et al. 1987).



Kuva 4.7. Esimerkki komponentista Grasshopper-ympäristössä.

Kuvassa 4.7 on esimerkki komponentista. Kuvan komponentti jakaa janan pituuden annettuun määrään osia, joiden pituus lineaarisesti riippuvainen toisistaan. Komponentin syötteeksi annetaan jana, haluttu välipistemäärä, sekä vakioarvo (lineaarisen funktion kulmakerroin). Tuloksena saadaan kunkin jakopisteen etäisyys janan alkupisteestä. Grasshopper-ympäristössä komponentti on siis muodostettu tarpeellisesta määrästä ympäristön tarjoamista alkeiskomponenteista jonkin tehtävän suorittamiseksi. Tällä tavoin muodostettuja komponentteja on helppo hyödyntää seuraavissa malleissa tarpeen mukaan.

Gamma (1995) esittää neljä kohtaa, jotka ohjelmistosuunnittelumallilta tulee löytyä. Mallilla tarkoitetaan tässä tehtäväsarjaa, ikään kuin algoritmia, jota noudattamalla jokin ongelma voidaan ratkaista. Vaikka kohdat on tarkoitettu uudelleenkäytettäväksi ohjelmistosuunnittelun toimintamallin pohjaksi, voidaan niitä soveltaa myös edellä esitettyihin komponentteihin. Soveltaen Gammaa (1995), komponentilla tulee siis olla:

1. nimi
2. ongelma
3. ratkaisu
4. seuraukset.

Komponentilla tulee olla nimi. Tämä luo pohjan komponentin uudelleenkäytettävyydelle ja dokumentoinnille. Se kertoo käyttäjälle heti, minkä tyyppiseen tilanteeseen komponentti on sovellettavissa. Ongelmalla tarkoitetaan tilannetta, johon komponentti on sovellettavissa. Toisin sanoen ilman ongelmaa ei tarvita myöskään komponenttia. Ratkaisu on edellä mainitun ongelman ratkaisu eli komponentin toiminnallisuus. Viimeisenä kohtana komponentilta löytyy aiemmista kohdista johtuen seuraukset. Hyödyntämällä komponenttia saadaan jonkinlainen lopputulos (output). Tämä on komponentin tavoiteltu seuraus. On kuitenkin hyvä huomata, että lopputuote riippuu komponentin ratkaisusta. Riippuen siitä mihin lopputulosta hyödynnetään, saattaa esimerkiksi sen muoto olla sellainen, joka rajoittaa sovellettavuutta parametrinen mallin rakentamisen myöhemmissä vaiheissa ja edelleen johtaa lisätyöhön.

Jos uudelleenhyödyntäminen lyhentää mallin luomiseen kuluvaa aikaa, parantaa se myös mallin laatua (Lim 1994). Tämä johtuu siitä, että uudelleenhyödynnettävät osat ovat jo vähintään kertaalleen käytettyjä ja testattuja. Näin ollen niiden toiminta suunnitellulla tavalla on varmennettu. Lisäksi mahdollisten virheiden korjaukset kumuloituvat, kun samaa osaa hyödynnetään uudelleen (Lim 1994). Edelleen, jos uudelleenhyödyntämiseen soveltuva materiaali on dokumentoitu sen käyttötarkoituksen osalta, pienentää se riskiä soveltamisesta väärään tai soveltumattomaan käyttötarkoitukseen.

Kokonaisen algoritmin tai parametrinen mallin uudelleenhyödyntäminen edellyttää koodin selventämistä kommenteilla. Perinteisissä ohjelmointikielissä käytetään tyypillisesti kommentteja koodin eri osien kuvaamiseen ja selventämiseen siten, että myös ulkopuoliset käyttäjät voivat saada selvää siitä, mitä kukin osa tekee. Visuaalisissa ohjelmointimenetelmissä ei tällaista ominaisuutta ole, mutta asia on helposti ratkaistavissa tekstisyötteiden antamiseen tarkoitetuilla ohjelmiston alkeiskomponenteilla. Alkeiskomponentteihin voidaan kirjoittaa selventäviä tekstejä ja kommentteja, joilla mallin toimintaa voidaan selventää ulkopuoliselle käyttäjälle. Näin voidaan pyrkiä välttämään edellä esitettyä ongelmaa siitä, että ainoastaan alkuperäinen tekijä ymmärtää luomaansa parametrista mallia. Käytännössä myös tämä edellyttää, että parametrinen mallin tekijä käyttää tietoisesti resursseja siihen, että malli olisi mahdollisimman helposti uudelleenhyödynnettävissä.

Uudelleenhyödyntämisen mahdollistamiseksi tarvitaan myös menetelmä, jolla materiaali on löydettävissä (Aberdeen 2007). Tähän vaikuttaa olennaisesti materiaalin määrä. Pienellä määrällä materiaalia, voidaan uudelleenhyödynnettävyys toteuttaa esimerkiksi säilömällä materiaali kansioon, johon kaikilla käyttäjillä on pääsy. Kun materiaalin määrä kasvaa suuremmaksi, tarvitaan tietokantatyyppinen ratkaisu, josta sopiva materiaali on löydettävissä esimerkiksi hakutoiminnolla. Lisäksi tarvitaan dokumentointijärjestelmä,

jolla esimerkiksi komponenttien ominaisuudet ja hyödynnettävyys voidaan tuoda ilmi. Materiaalin arkistointijärjestelmän ja dokumentointijärjestelmän tulee myös yhdessä mahdollistaa löytyvien virheiden korjaus siten, että käyttäjien saatavilla on aina uusin versio, vaikka vanhat versiot onkin syytä arkistoida mahdollisia tulevia kehitystarpeita varten. Näiden järjestelmien toteuttaminen on kuitenkin tietotekninen tehtävä, tässä on esitetty ainoastaan edellytykset järjestelmän toiminnalle.

Toisaalta tehokkuus voi perustua juuri suunnitelman muuttamisen tehokkuuteen, kuten luvussa 4.3.1 on esitetty. Tällöin avainasemassa ei ole se, miten tehokkaasti parametrinen malli on luotu, vaan se miten paljon hukkaan mennyttä työtä muutosten toteuttaminen aiheuttaisi, mikäli muutokset toteutettaisiin perinteisillä suunnittelumenetelmillä. Yksinkertaistetusti voidaan sanoa, että mikäli luonnossuunnitelman sekä muutosten toteuttamiseen perinteisillä suunnittelumenetelmillä kuluu enemmän aikaa kuin parametrisen mallin luomiseen ja muutosten toteuttamisiin sillä, on parametrinen suunnittelu tehokasta. Ongelmaksi muodostuu, ettei työmäärää voida eksplisiittisesti tietää ennen suunnittelutyön valmistumista. Lisäksi suunnittelutyön valmistuttua tiedetään ainoastaan valittuun ratkaisuun käytetty työmäärä.

On huomattava, että mikäli suunnitteluratkaisu muuttuu perustavanlaatuisesti, on luotu parametrinen malli tämän suunnittelukohteen kannalta hyödytön (Smith 2007). Toisin sanoen mallintaessaan parametrisesti suunnittelija valitsee parametreja ja reunaehtoja, joita sisällytetään malliin. Samalla rajataan suunnitteluavaruutta, josta lopullinen suunnitelma voidaan valita (parametrisen mallin avulla). Perustavanlainen muutos, jossa parametrinen malli muuttuu hyödyttömäksi, on siis sellainen, jonka johdosta tavoiteltava ratkaisu ei löydy rajatun suunnitteluavaruuden sisältä. Tällaisia tilanteita ovat esimerkiksi kun arkitekhti muuttaa rakennuksen massoittelua tai kun suunnitteluratkaisua muutetaan käytettävän rakenneosan osalta.

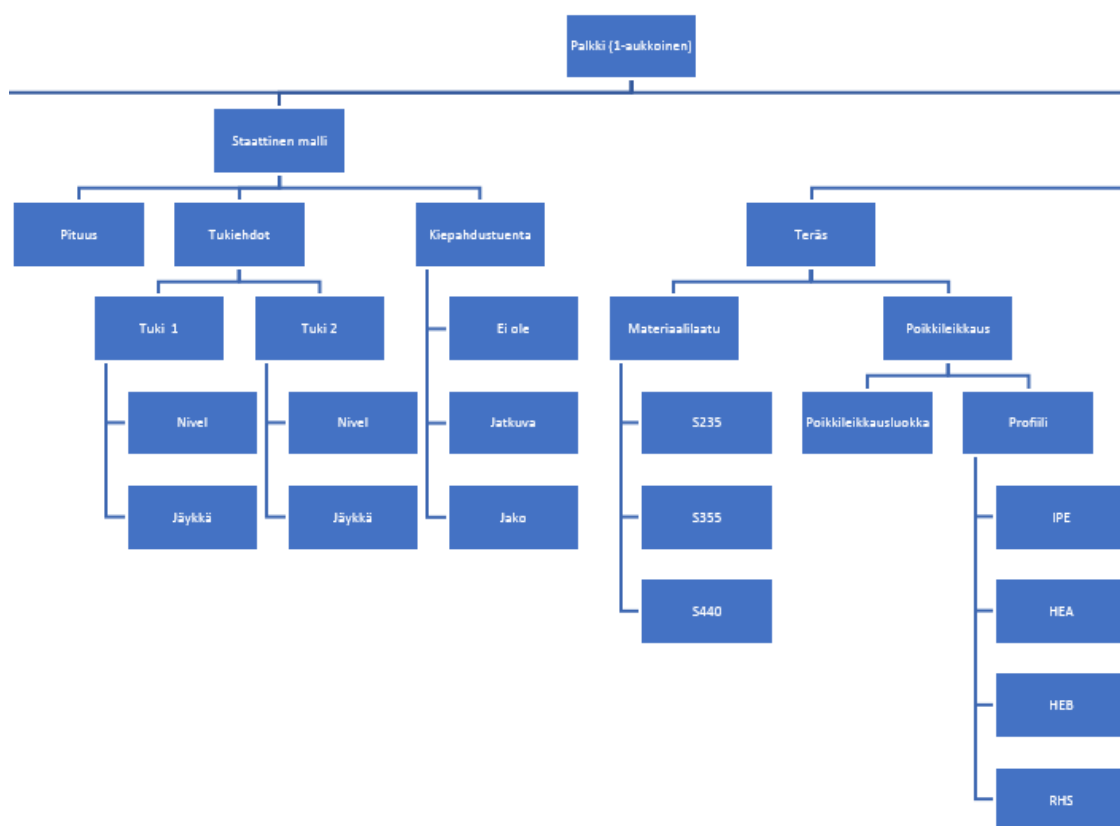
Mikäli tällainen muutos tehdään, on sillä merkittävän negatiivinen vaikutus parametrisen suunnittelun tehokkuudelle. Tämä siksi, että parametrisen mallin toimivaksi saattaminen vaatii mahdollisesti niin merkittäviä muutoksia, että voi olla järkevämpää aloittaa puhtaalta pöydältä tai viedä suunnittelu loppuun käyttäen perinteisiä menetelmiä. Tämän ollessa tilanne, kaikki parametrisen mallin luomiseen käytetty työ on hukkaa, pois lukien tulevaisuudessa mahdollisesti uudelleenhyödynnettävät osat.

Edellä mainitut yhdessä määräävät sen, onko parametrinen mallintaminen tehokasta. Uudelleenkäyttämällä jo luotuja parametrisen mallin osia ja komponentteja lyhennetään mallin luomiseen kuluva aikaa ja parannetaan mallin laatua (Lim 1994). Tällä tavoin voidaan säästää merkittävä määrä aikaa parametrisen mallin luonnissa sekä luoda laadukkaampi malli. Tehokkuuden määrää kuitenkin se, onko suunnitelman luominen parametrisesti ollut nopeampaa kuin perinteisillä menetelmillä, kun tilannetta tarkastellaan projektin päätteeksi. Tämän toteaminen yksiselitteisesti on vaikeaa. Tilannetta voidaan kui-

tenkin arvioida keräämällä organisaation sisällä tietoa suunnitelmien luomiseen käytetystä ajasta molemmilla menetelmillä. Tähän tietoon sekä projektin tunnuslukuihin verraten voidaan tällöin pyrkiä tekemään päätelmiä parametriseen mallintamiseen kuluva ajasta ja sen tehokkuudesta.

4.4 Rakenteen mallintaminen

Rakenteen mallintaminen on aloitettava valitsemalla parametrit, joita halutaan käyttää. Valitsemista varten voidaan määritellä parametripuu. Parametripuussa esitetään asiat, joista rakenneosan rakennetekninen toiminta muodostuu. Tällaisia asioita ovat geometrian lisäksi esimerkiksi materiaali, kuormitus sekä tukiehdot.



Kuva 4.8. Esimerkki parametripuun osasta.

Kuvassa 4.8 on esitetty osa 1-aukkoisen palkin parametripuusta ja siinä näkyy osa palkin valintaan vaikuttavista tekijöistä. Rakennesuunnittelussa parametrit linkittyvät toisiinsa suunnitteluohjeiden kautta, koska rakenteiden on täytettävä geometrysten vaatimusten lisäksi toiminnalliset eli rakennetekniset ja taloudelliset vaatimukset. Edelleen eri sidosryhmät, kuten arkkitehti tai tilaaja, voivat lukita osan parametreista. Esimerkiksi tilaaja saattaa haluta määrittää rakennusmateriaalin ja arkkitehti antaa tietyn tilavaruksen rakenneosalle.

Erityisesti esisuunnitteluvaiheessa, kun rakennuksen rakennemallia vasta määritellään, on parametripuuta syytä tutkia ylemmältä tasolta. Esimerkiksi yksiaukkoisen palkin asemesta voidaan tarkastella erilaisia vaihtoehtoja, kuten ristikkoja tai jatkuvia palkkeja. Edelleen kaikkien rakenneosien alla on niihin vaikuttavat parametrit, jotka lopputilanteessa määrittävät rakenneosan soveltuvuuden. Voidaan todeta, että koko rakennuksen kantava runko (tai jopa koko rakennus) on teoriassa mahdollista parametrisoida. Rajoitteita tähän luo käytettävissä olevan välineistön ja suunnitteluohjelmiston kapasiteetti ja soveltuvuus sekä käytettävissä oleva aika.

Toisaalta tulee pystyä määrittämään mikä on parametrisen suunnittelun tavoite. Parametrisilla menetelmillä kyetään esimerkiksi tutkimaan erilaisia vaihtoehtoja rakennuksen rungon rakennemalleille, arkkitehdin suunnitelman toteutusvaihtoehtoja tai pyrkiä etsimään mahdollisimman taloudellista ratkaisua johonkin suunnitteluongelmaan. Tavoitteen määrittäminen luo pohjan mallintamiselle ja parametrien määrittämiselle. Kuten luvussa 4.3 on esitetty, parametrisen mallintamisen tehokkuus perustuu siihen, että viimekädessä käytetty aika on perinteistä suunnittelumenetelmää pienempi. Mikäli tavoitetta ei osata määrittää tai se määritetään väljästi, johtaa tämä siihen, että parametrasta mallia luodessa tulee kaikki mahdolliset muutoksia aiheuttavat tekijät ottaa huomioon. Tämä johtaa ensinnä mallin käytettävän työmäärän merkittävään kasvamiseen. Toiseksi, ilman selkeää tavoitetta olennaisten asioiden tunnistaminen on vaikeaa, mikä saattaa johtaa siihen, ettei niitä oteta huomioon mallia rakennettaessa. Tämä johtaa pahimmillaan luvussa 4.3.2 esitettyyn tilanteeseen, jossa malli ei sovellu käyttötarkoitukseen ja on näin ollen hyödytön.

Näin ollen valinta mallinnettavasta rakenneosasta on tehtävä mallin suunnittelun yhteydessä. Malliin voidaan tietysti sisällyttää erilaisia vaihtoehtoja (esimerkiksi vaakakannattajille yksiaukkoisia ja jatkuvia palkkeja sekä ristikoita), mutta tämä lisää mallin luomiseen kuluva aikaa merkittävästi. Tästä johtuen tällaisen mallintamisen on oltava perusteltua ja se on otettava huomioon mallin suunnitteluvaiheessa. Rakennesuunnittelulle on melko tyypillistä, että arkkitehti sitoo vähintään osan geometriasta. Tämä rajaa osaltaan käytettävissä olevia rakennevaihtoja, eli helpottaa valintaa käytettävän rakennerratkaisun suhteen ja edelleen mallin rakentamisen suhteen. Toisaalta tämä pitää sisällään riskin siitä, että arkkitehdin tekemät merkittävät muutokset tekevät rakennesuunnittelijan mallin käyttökelvottomaksi.

Tehokkaan mallintamisen takaamiseksi rakenneosien mallintamisessa on pyrittävä hyödyntämään mahdollisimman paljon aikaisemmin luotuja komponentteja ja algoritmeja. Tämä sopii rakennesuunnitteluun hyvin: Lenz et al. (1987) vertauksen mukaan komponentit ovat kuin rakenneosia. Sovellettaessa parametrasta suunnittelua rakennesuunnitteluun, voidaan ajatuksesta johtaa seuraava: rakenneosat ovat komponentteja. Kun tällä ajatuksella luodaan lukuun 4.3.2 perustuen komponentteja, joiden avulla voidaan parametrisesti mallintaa rakenneosia, voidaan parametrinen mallintaminen toteuttaa tehokkaasti. Tätä on käsitelty tarkemmin luvuissa 5.3 ja 5.7.

Rakenteen mallintamisessa on siis otettava huomioon arkkitehdin määräämä geometria, jota voidaan parhaimmillaan käyttää myös parametrisen mallin pohjana. Lisäksi on otettava huomioon muista tekijöistä aiheutuvat reunaehdot. Näiden sekä parametrisen mallin käyttötarkoituksen ja tavoitteiden perusteella tehdään päätös mallinnettavasta rakenneosasta tai -osista. Rakenneosat mallinnetaan hyödyntämällä soveltuvien osien olemassa olevia komponentteja. Näin mallin rakentaminen on tehokasta, malli vastaa käyttötarkoitusta ja se on toiminnallisesti laadukas.

5. PARAMETRISOINNIN HYÖDYNTÄMINEN RAKENNESUUNNITTELUSSA

5.1 Parametrinen mallintaminen

Tärkein lähtökohta parametriseen mallintamiseen ja suunnitteluun on taloudellisuus. Rakennuksen tai rakenteen parametrinen malli on teollisissa ja kaupallisissa sovelluksissa luotava siten, että siitä saadaan jotain hyötyä verrattuna perinteisiin suunnittelumenetelmiin. Käytännössä tämän hyödyn on viimekädessä tarjottava kustannushyötyjä joko tilaajalle tai suunnittelijalle, jotta on järkevää suunnitella parametrisesti. Yksinkertaistettuna kustannushyötyjä saavutetaan tilaajan kannalta, kun rakennus on edullisempi toteuttaa. Tällöin suunnittelijan on kyettävä osoittamaan jo suunnittelutarjousvaiheessa, että kykenee tarjoamaan kilpailijoita edullisemman suunnitteluratkaisun.

Rakennesuunnittelijan kannalta kustannushyödyt taas tarkoittavat lyhyempää suunnittelu-aikaa. Vaikka tässä raportissa käsitellään rakenteen optimointia ja rakenteen parametrisointia siten, että optimointi on mahdollista, keskitytään kuitenkin esittämään parametrinen suunnittelu rakennesuunnittelijan työtä helpottavana ja nopeuttavana työkaluna. Tämä siksi, että suunnitteluratkaisun edullisuuden osoittaminen tilaajalle, erityisesti verrattuna kilpailijan ratkaisuun, on vaikeaa. Tapaukset, joissa parametrisella suunnittelumenetelmällä saavutetaan hyötyä rakennesuunnittelijan näkökulmasta, on esitetty luvussa 4.3.

5.2 Parametrisen mallin suunnittelu

Parametrinen malli tulee suunnitella ennen sen luontia. Tämä johtuu siitä, että parametrista mallia luotaessa tulee ensin muodostaa ajatus siitä mitä on tekemässä ja mitä sen saavuttamiseksi vaaditaan (Smith 2007). Suunnittelemalla varmistetaan, että:

- malli vastaa tarkoitustaan
- malli on taloudellinen toteuttaa
- mallin tuottama tieto soveltuu jatkokäsittelyyn
- malli on soveltuvilta osin hyödynnettävissä tulevilla kohteilla.

Jotta yllä esitettyihin tavoitteisiin päästään tulee ensin määritellä mallin käyttötarkoitus eli mitä parametrisella mallilla halutaan saavuttaa. Tyypillisesti käyttötarkoitus perustuu johonkin luvussa 4.3 esitetystä hyötynäkökohdasta. Näihin perustuen mallin käyttötarkoitus voi olla esimerkiksi:

- useiden suunnitteluratkaisujen vertailu (luonnossuunnitteluvaiheessa)
- muutosten aiheuttaman työmäärän minimointi
- monimutkaisen geometrian luominen.

Kun mallin käyttötarkoitus on määritetty, voidaan määrittellä tavoitteet parametrille mallintamiselle. Tavoitteet perustuvat mallin käyttötarkoitukseen. Tavoitteet voidaan asettaa esimerkiksi kysymyksellä: ”Mitä ominaisuuksia mallin pitää toteuttaa, jotta käyttötarkoitus saavutetaan?” Tavoitteilla siis ohjataan mallin luontia, jotta käyttötarkoitus voidaan saavuttaa mahdollisimman hyvin.

Kun edellä mainitut asiat on määritetty, voidaan mallin ominaisuudet suunnitella. Käytännössä suunnittelu alkaa käytettävien ohjelmistojen valinnalla. Kun parametriseen mallintamiseen käytettävä ohjelmisto on valittu, voidaan tavoitteisiin ja käyttötarkoitukseen perustuen määrittellä mitä lisäosia ja muita ohjelmistoja suunnittelutyössä käytetään. Tällaisia ovat esimerkiksi rakenneanalyysiin käytettävät FEM-ohjelmistot sekä suunnitelmien tuottamiseen käytettävä tietomallinnusohjelmistot. Ohjelmistojen määrittämisen lisäksi tulee suunnitella miten tiedonsiirto eri ohjelmien välillä toteutetaan.

Kun esitetty suunnittelun ja mallintamisen perusedellytykset on määritetty, voidaan määrittää mallin olennaiset reunaehdot ja parametrit. Reunaehdot määräytyvät yleensä pääosin tilaajan ja arkkitehdin määritelmien perusteella. Riippuen mallista reunaehdot voivat olla mitä tahansa geometriasta esimerkiksi palonkestovaatimuksiin. Mallin käyttötarkoitus määrää miten ja mitkä reunaehdot otetaan huomioon mallia luodessa. Vastaavasti myös parametrit ja niiden valinta riippuvat mallin käyttötarkoituksesta kuten luvussa 2.5 on esitetty. Suunnittelemalla varmistetaan, että parametrien määrä vastaa mahdollisimman täsmällisesti käyttötarkoitusta: liian pienellä parametrimäärällä mallille asetettuja tavoitteita ei välttämättä saavuteta ja toisaalta liian suuri parametrimäärä johtaa suureen työmäärään.

Parametrin mallin suunnitteluvaiheessa tulee ottaa huomioon yhteistyö muiden suunnittelualojen kanssa sekä olevien ja tulevien komponenttien ja algoritmien hyödyntäminen. Yhteistyötä muiden suunnittelualojen kanssa on käsitelty luvussa 5.5. Lisäksi mallin suunnittelussa on käsiteltävä uudelleen hyödyntämiseen liittyvät näkökulmat. Nämä on käsitelty luvussa 4.3.2.

5.3 Rakenneosien mallintaminen parametrisesti

5.3.1 Rakenneosan parametrisointi

Yleisesti voidaan todeta, että parametrinen mallintaminen soveltuu kaikkiin kohteisiin, ainoana rajoitteena parametrin menetelmien soveltamiseen ovat suunnittelijan osaaminen, käytettävissä olevat ohjelmistot sekä käytettävissä oleva laskentateho. Parametriseen suunnitteluun liittyy kuitenkin mallintamisen lisäksi myös rakenteen toiminnallisuuden

arviointi. Kantavien rakenteiden osalta tämä tarkoittaa lujuuden ja muodonmuutosten laskentaa ja varmistumista siitä, että pysytään suunnittelunormien määrittämien raja-arvojen sisällä.

Luvun 2 perusteella rakenteen soveltuvuus voidaan varmistaa joko yksinkertaisella normiin perustuvalla laskentamallilla parametrisessa suunnitteluohjelmassa tai vaihtoehtoisesti suunnitteluohjelmaan linkitettävällä FEM-laskennalla. Menetelmien rajoittuneisuus rajoittaa tällä hetkellä myös parametrinen suunnittelun sovellettavuutta.

Yksinkertaisin sovelluskohde parametriselle suunnittelulle ovat sauvamaiset rakenteet ja niiden muodostamat kokonaisuudet kuten pilarit, palkit ja ristikot, joiden kestävyyttä ja siirtymiä voidaan arvioida tukiehtojen, pituuden ja poikkileikkausvakioiden avulla. Tämä johtuu siitä, että jana on graafisten suunnittelumenetelmien peruselementti. Janalla on pituus ja sille voidaan parametrisesti määrittää tukiehdot ja poikkileikkausarvot, joiden avulla siis sen kapasiteetti voidaan määrittää. Lisäksi laskelmat ovat statiikasta riippuen yleensä melko yksinkertaisia. Sauvamaiset rakenteet on myös helppo tuoda FEM-analyyysiin, jolla myös statiikaltaan monimutkaisten rakenteiden analyysi onnistuu.

Muiden rakenneosien osalta parametrinen suunnittelun soveltaminen vaatii laajempaa osaamista johtuen laajemmasta parametrimäärästä ja suunnitteluratkaisun soveltuvuuden vaikeammasta arvioinnista. Käytännössä tämä tarkoittaa laajempia ja työlämpiä parametrisia malleja.

Parametrinen mallintaminen ja suunnittelu on myös materiaaliiriippumatonta, rajoittavina tekijöinä ovat edelleen osaaminen ja laskentateho. Homogeenisena ja isotrooppisena materiaalina teräs on yksinkertaisin materiaali mallintaa parametrisesti. Tämä johtuu siitä, että tavallisissa tapauksissa teräksen lujuus riippuu ainoastaan sen lujuusluokasta. Paksujen teräsosien tapauksessa tulee ottaa huomioon myös ainevahvuus. Näiden kahden parametrien avulla voidaan ratkaista suurin osa rakenneteknisistä suunnittelutehtävistä. Teräkseen verrattuna puu on monimutkaisempi mallintaa rakenneteknistä näkökulmasta, koska sen lujuuteen vaikuttaa kuormitustilanne sekä rakenneosan geometria. Tämä kasvattaa tarvittavien parametrien määrää merkittävästi, mikä edelleen vaikeuttaa parametrilla mallintamista. Kolmesta merkittävästä rakennusmateriaalista haastavimpana voidaan pitää betonia. Tämä johtuu siitä, että betonipoikkileikkauksen kapasiteettiin vaikuttaa merkittävästi rauditusgeometria. Lisäksi jännitysten jakautumiseen vaikuttaa kuormitustapa, jolloin kapasiteettitarkistuksen parametrimäärä kasvaa merkittävästi.

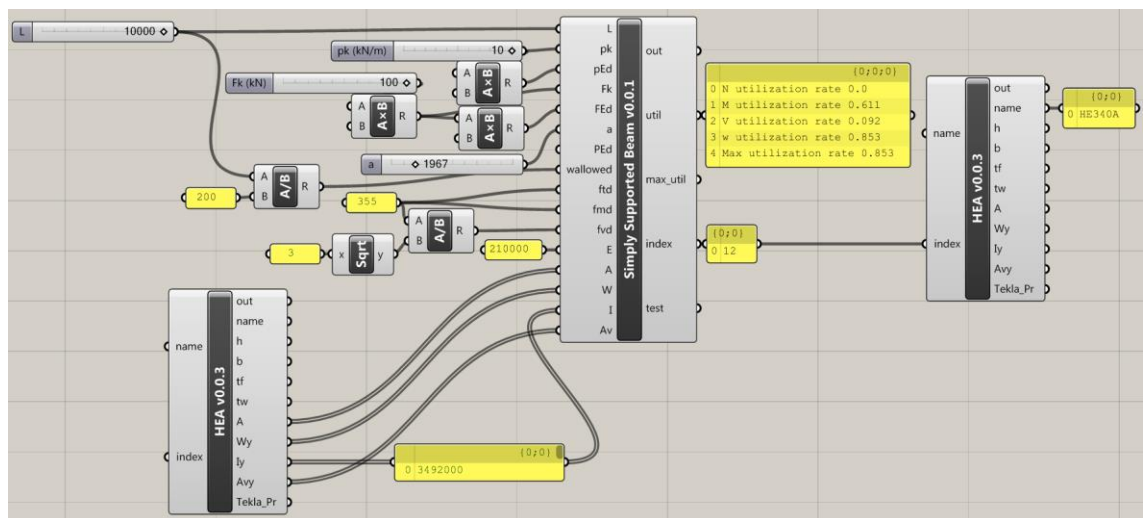
Materiaalikohtaisia haasteita voidaan toisaalta tarkastella myös mahdollisuuksina. Kuten kuvassa 4.2 on esitetty, suunnittelussa on käytävä läpi samat vaiheet riippumatta, siitä tehdäänkö se perinteisesti vai parametrisoidusti. Näin ollen automatisoimalla haastavampia suunnittelutehtäviä parametrisilla menetelmillä, voidaan suunnitteluun käytettävää aikaa lyhentää verrattuna perinteiseen menetelmään. Haastavissa tehtävissä, joihin perin-

teisillä menetelmillä kuluu suurempi aika, on ajallinen hyöty suurempi, koska automaattisen prosessin läpivieminen on merkittävän lyhyt eikä se juurikaan riipu tehtävän haastavuudesta, mikäli tarvittavat komponentit ovat olemassa.

5.3.2 Palkki

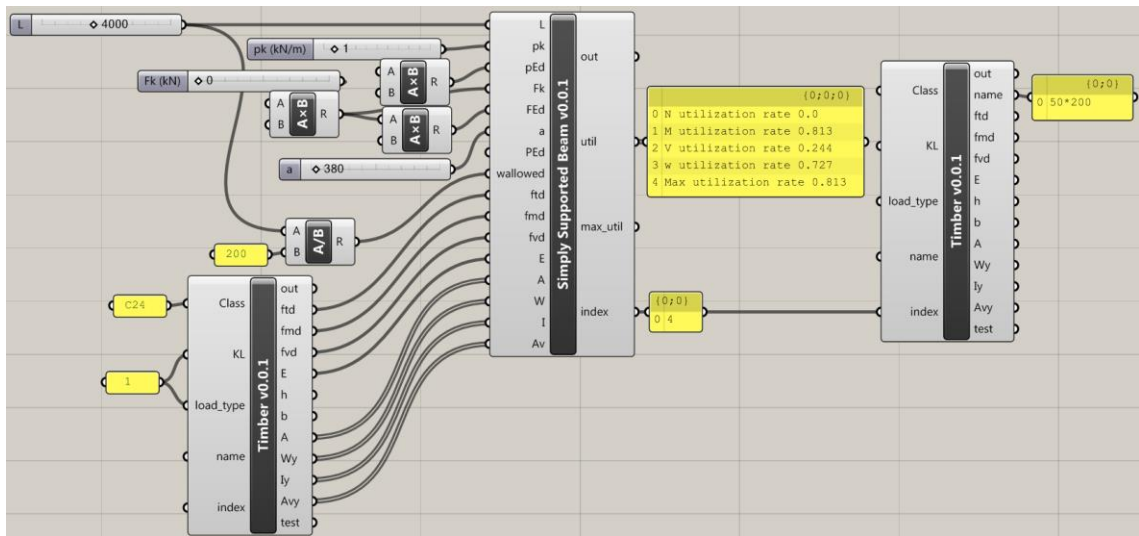
Palkin mallinnuksessa geometrian peruselementti on jana, joka määrää palkin pituuden. Jotta palkin rakenteellista toimintaa voidaan arvioida, on sille annettava myös poikkileikkaus, lujuustekniset ominaisuudet sekä tukiehdot. Näiden perusteella voidaan arvioida palkin kuormankantokykyä. Yhdessä nämä tekijät muodostavat palkin parametripuun. Kuten aiemmin on esitetty, parametripuun avulla valitaan rakenneosan parametrit ja reunaehdot. Valitut parametrit mahdollistavat parametrin mallin kattavan suuremman osan suunnitteluavaruutta, kun taas reunaehdot rajoittavat sitä.

Suunnittelijan on helppo ymmärtää kaikki palkin toimintaan vaikuttavat tekijät ja luoda niiden perusteella parametrinen rakenneosia. Tässä työssä osoitettiin palkin soveltuvuus parametriseen mallintamiseen luomalla siitä uudelleenhyödynnettävä parametrinen komponentti Grasshopper-ympäristössä. Komponentti on esitetty kuvassa 5.1.



Kuva 5.1. Parametrinen palkkikomponentti.

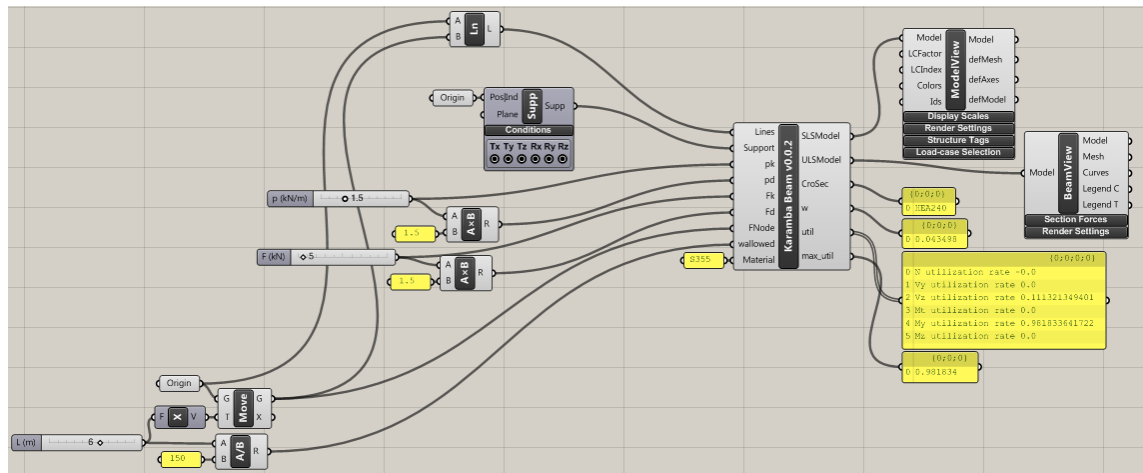
Kuvan 5.1 analyttisessä komponentissa palkin parametreiksi on valittu ulkoisten kuormien ominais- ja suunnitteluarvot, sallittu taipuma, lujuusominaisuudet sekä poikkileikkauksen kapasiteettiin vaikuttavat tiedot. Näillä parametreilla saavutetaan suhteellisen vapaa kuormitustilanne (yksi viivakuorma, yksi pistekuorma vapaassa sijainnissa sekä normaalivoima) sekä mahdollisuus käyttää teräs- ja puupoikkileikkauksia. Komponentin käyttökelpoisuuden ja yksinkertaistamisen vuoksi osa parametreista on kuitenkin lukittu: komponentti mitoittaa ainoastaan yksiaukkoisen, vapaasti tuetun palkin eikä sillä voi mitoittaa betonipoikkileikkausta. Vertaamalla kuvaa 5.1 kuvaan 5.2 havaitaan, että kahden eri materiaalin (teräs ja puu) mitoituksen mahdollistaminen kasvattaa tarvittavaa vapaiden parametrien määrää merkittävästi.



Kuva 5.2. Palkkikomponentti sahatavaran mitoituksessa.

Puun materiaaliominaisuudet vaihtelevat lujuusluokan, rasituksen sekä käyttöluokan ja kuorman aikaluokan perusteella. Teräkselle taas lujuuteen vaikuttavat yleensä ainoastaan lujuusluokka ja ainevahvuus. Jotta puupoikkileikkauksen mitoittaminen olisi mahdollista, tulee siis vapauttaa parametreja, jotka teräkselle ovat riippuvia muista parametreista.

Analyttisen komponentin vaihtoehtona voidaan käyttää FEM-pohjaista komponenttia. FEM-komponentti vaatii käytettävän ympäristön kanssa yhteensopivan FEM-ohjelmiston, jotta tietoa on mahdollista kuljettaa eri ohjelmistojen välillä. Grasshopper-ympäristössä toimii esimerkiksi Karamba FEM-ohjelmisto (Preisinger 2016). Kirjoitushetkellä yhteensopivat FEM-ohjelmistot ovat melko karkeita, ne on tarkoitettu pääasiassa arkkitehtien käyttöön ja niistä puuttuu monia rakennesuunnittelussa käytettäviä ominaisuuksia. Ne kuitenkin soveltuvat hyvin juuri yksinkertaisten rakenteiden mitoittukseen, joskin tulokset on syytä varmistaa jollakin muulla menetelmällä ennen lopullisten suunnitelmien tekoa.



Kuva 5.3. FEM-pohjainen palkkikomponentti.

Kuvassa 5.3 on esitetty esimerkki Grasshopper-ympäristössä toteutetusta Karamba-komponentista. Kuvasta nähdään, että syöteparametrien määrä vastaa melko hyvin kuvassa 5.1 esitettyä analyyttistä palkkikomponenttia. Molemmissa komponentille annetaan lähtötiedoiksi palkin geometria (jännemitta), kuormitustiedot sekä sallittu taipuma. Selkein ero on, että kun analyyttiselle komponentille tukiehdot oli lukittu, ovat ne FEM-pohjaisessa komponentissa vapaita mahdollistaen esimerkiksi jatkuvien palkkien ja ulokepalkkien analyysin kuten kuvassa 5.4 on esitetty.



Kuva 5.4. FEM-komponentilla analysoitu ulokepalkki.

Vastaava toteutus olisi tietysti mahdollista myös analyyttisessä komponentissa siinä laajuudessa, kun analyyttiset ratkaisut tunnetaan. Erilaisten tapausten lisääminen on kuitenkin merkittävästi työläämpi, sillä komponentille pitäisi ohjelmoida erilaiset ratkaisut sekä

algoritmi, joka osaa valita, mitä ratkaisua käytetään. Toinen merkittävä ero on FEM-ohjelmiston valmiin materiaalikirjaston hyödyntäminen FEM-komponentissa. Tässä tapauksessa parametreina annetaan ainoastaan materiaalin nimi ja ohjelma valitsee kirjastosta automaattisesti tarvittavat materiaalitiedot ja lujuusarvot. Vastaava kirjastopohjainen toteutus on hyvä lähestymistapa myös analyttisessä komponentissa. Voidaan kuitenkin todeta, että koska visuaalisista ohjelmointiympäristöistä puuttuvat tällä hetkellä suuri osa jokapäiväisessä rakennesuunnittelussa tarvittavista työkaluista, on komponenttien ja työkalujen luojalla vapaus päättää työkalujen hierarkiarakenteesta. Näin ollen materiaalikirjasto voidaan toteuttaa täysin erillisinä, yhteensopivina komponentteina tai yhdistää esimerkiksi poikkileikkauskirjastoihin kuten kuvissa 5.1 ja 5.2 on tehty.

Myös kolmas komponenttien välinen ero on kirjastopohjainen. Analyttisessä komponentissa komponentille on annettava parametrina poikkileikkaukset, joista optimaalinen ratkaisu valitaan. FEM-pohjaisessa komponentissa voidaan jälleen hyödyntää valmista profiilikirjastoa, josta poikkileikkaus valitaan. Valmiin kirjaston etuna saavutetaan laaja lukitsemattomien poikkileikkausten määrä. Haittapuolena poikkileikkauksen tyyppiä (esimerkiksi IPE tai HEA) ei voida valita ilman lisäparametreja.

5.3.3 Pilari

Geometriselta olemukseltaan pilari vastaa edellisessä luvussa esitettyä palkkia, joten sen parametrisoiminen on luontevasti palkin jälkeen seuraava askel. Pilari on kuitenkin palkkia monimutkaisempi rakenne, johtuen sille tyypillisestä puristuskuormituksesta, jonka johdosta pilarien rakenneanalyysissä tulee ottaa taivutusmomenttia aiheuttava puristavan kuorman epäkeskeisyys (niin todellinen kuin teoreettista epätarkkuuksistaikin johtuva). Lisäksi puristetuille rakenteille tulee tyypillisesti tarkastella nurjahdustilanne. Eurokoodimitoituksessa pilarin epätarkkuudet voidaan ottaa huomioon poikittaisvoimalla, joka riippuu alkuvinoudesta tai alkusivusiirtymästä. Materiaalikohtaisista eurokoodeista (SFS-EN 1992-1-1 2004; SFS-EN 1993-1-1 2005) yhdistämällä saadaan parametriseen suunnitteluun sopiva vinouden φ_i kaava

$$\varphi_i = \varphi_0 \alpha_h \alpha_m, \quad (12)$$

missä φ_0 on perusarvo $1/200$, α_h on pituuden tai korkeuden pienennystekijä ja α_m on rakenneosien määrään perustuva pienennystekijä. Korkeuden pienennystekijä α_h lasketaan kaavasta

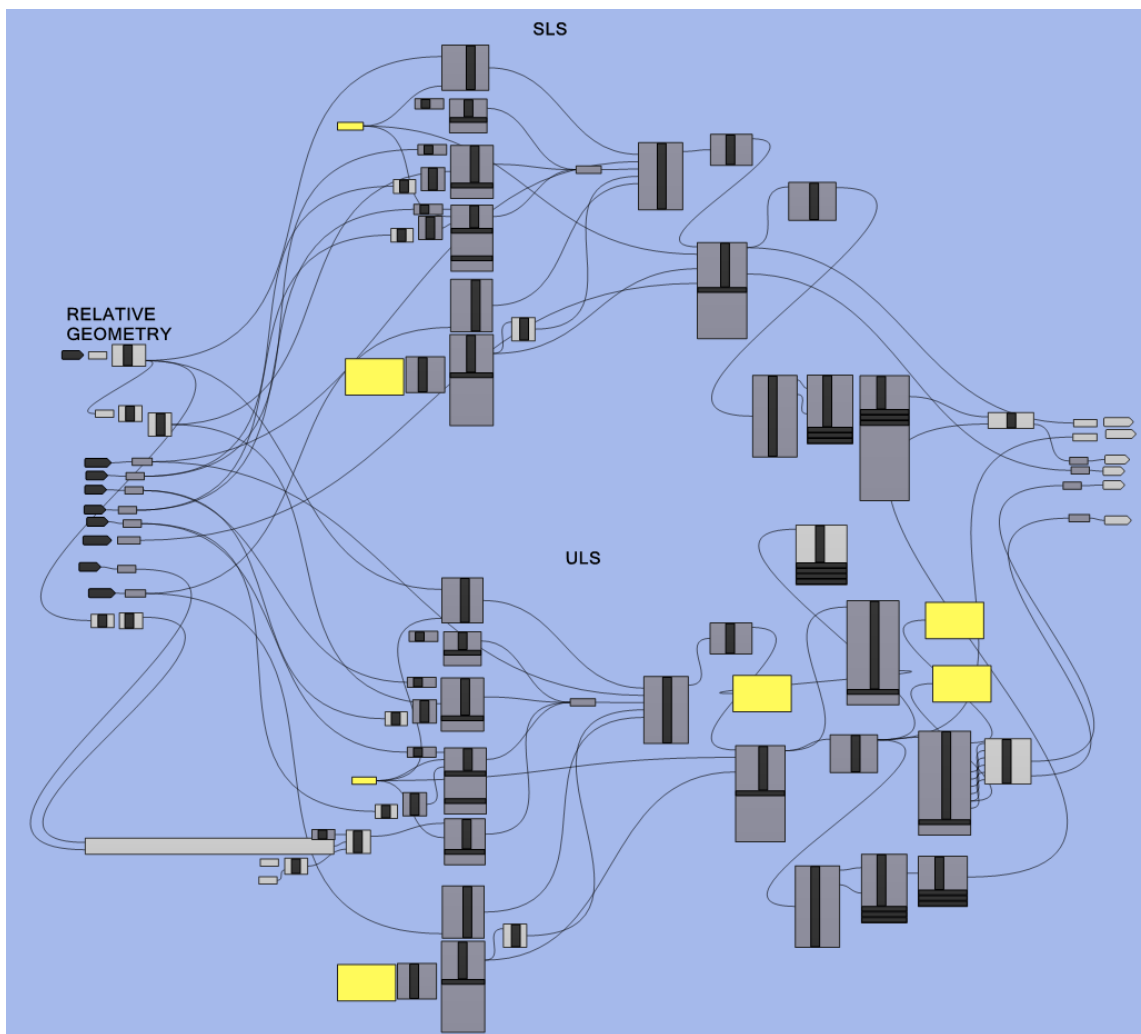
$$\alpha_h = \max\left(\min\left(2/\sqrt{h}, 1\right), \frac{2}{3}\right), \quad (13)$$

missä h rakenneosan korkeus tai pituus. Määrään perustuva pienennyskerroin α_m lasketaan kaavasta

$$\alpha_m = \sqrt{0,5 \left(1 + \frac{1}{m}\right)}, \quad (14)$$

missä m on kokonaisvaikutuksen aiheuttavien rakenneosien määrä. Poikittaisvaikutukset huomioon ottava poikittaisvoima saadaan kertomalla kuormitustilanteessa vaikuttava pystykuorma φ_i arvolla. Tämän kuorma asetetaan vaikuttamaan jäykistämättömän rakenneosan päihin siten että se aiheuttaa lisämomentin.

Pilarin soveltuvuutta parametriseen suunnitteluun tarkasteltiin FEM-komponentin avulla. Työssä onnistuttiin luomaan parametrinen malli, jossa pilarin lujuuslaskenta suoritettiin sen todellisessa geometrisessa sijainnissa parametrisessa mallissa. Komponentti on esitetty kuvassa 5.5.



Kuva 5.5. Pilarikomponentin toiminta.

Kuvassa 5.5 esitetty komponentti laskee teräspilarin siirtymän käyttörajatilassa sekä suorittaa kestävyystarkastelun murtorajatilassa. Komponentille on vapautettu kuormaparametreina pistekuorma pilarin päähän sekä tasainen kuorma, joka taivuttaa pilaria vahvemmassa suunnassa. Komponentti laskee pilarin epätarkkuudet edellä esitettyjen kaavojen mukaan sekä niitä vastaavat ekvivalentit voimat.

Ongelmalliseksi muodostui pilarin tukiehtojen määrittely parametrisesti, jota työssä ei onnistuttu toteuttamaan. Tukiehtojen parametrinen määrittely on olennaista pilarikomponentin käyttökelpoisuuden kannalta, koska niillä on merkittävä vaikutus niin pilarin rasiitustilaan kuin nurjahduspituuksiinkin. Vaikka pilarin nurjahduspituus voidaan määrittellä parametrisesti riippuen määrätystä tukiehdoista, ei tämä onnistunut FEM-laskentamallin osalta. Tästä johtuen pilarin tukiehdot joudutaan määrittämään käsin. Tämä johtaa seuraaviin ongelmiin:

1. mallin muutosjoustavuus vähenee
2. komponentin sovellusmahdollisuudet ovat merkittävän rajalliset
3. käyttäjän tekemät virheet johtavat helposti ristiriitoihin tukiehtojen ja nurjahduspituuksien välillä.

Mallin muutosjoustavuus vähenee, koska tukiehdoista ei voida tehdä sidottuja parametreja. Tämä tarkoittaa, että mallia muutettaessa on tarkasteltava, pätevätkö pilarille asetut tukiehdot. Toisin sanoen tukiehdot voivat olla teoriassa lukittuja parametreja, jotka riippuvat muiden parametrien arvoista, mutta tämä riippuvuus on tehtävä käsin. Koska tukiehtoja ei voida muuttaa algoritmeilla, komponenttia voidaan soveltaa ainoastaan tilanteissa, joissa ennalta määrätty tukiehdot täyttyvät. Koska tukiehdot joudutaan määrittämään laskentaa varten käsin, kun taas nurjahduspituudet voidaan määrätä algoritmisesti, käyttäjän tekemät virheet johtavat ristiriitaisuuksiin näiden toisistaan riippuvien tekijöiden välillä.

Kaikki esitetyt ongelmat ovat seurausta käytetyn ohjelmiston rajoittuneisuudesta. Koska FEM-ohjelmisto ei salli tukiehtojen parametrista määrittämistä se joudutaan tekemään käsin. Työssä onnistuttiin toteuttamaan parametrisesti kaikki pilarin FEM-laskentaan tarvittavat lähtötiedot, mutta niitä ei onnistuttu muuttamaan FEM-ratkaisijan tukemaan muotoon. Näin ollen pilarin soveltumisesta komponenttipohjaiseen parametriseen mallintamiseen ei tutkimuksen perusteella voida sanoa mitään. Voidaan kuitenkin todeta, että on mahdollista toteuttaa parametrisia malleja, joissa pilarit ovat osa rakennetta. Kuitenkaan pelkän pilarin mallintaminen parametrisesti ei ole järkevää johtuen menetelmän rajoittuneisuudesta ja saavutettavien hyötyjen vähäisyydestä.

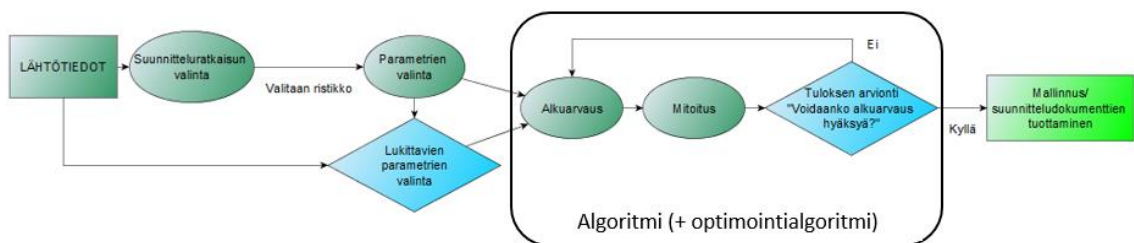
5.3.4 Ristikko

Verrattuna aiemmin esitettyihin rakenneosiin ristikon parametrinen suunnittelun ja mallintamisen hyödyt ovat selkeimmät. Ristikko koostuu useista sauvoista, paarteista ja diagonaaleista, joiden sijoittelu määrää rakenteen kuormankantokapasiteetin. Teräsristikkoille on esitetty omat suunnitteluohjeensa lähteissä (SFS-EN 1993-1-1 2005; SFS-EN 1993-1-8 2005). Puisten naulalevyristikoiden osalta eurokoodi 5:en (SFS-EN 1995-1-1 + AC 2006) pohjautuva (RIL 205-1-2017 2017) käsittelee lyhyesti ristikoiden staattista mallia ja toteaa, että naulalevyliitokset on suunniteltava erillisen ohjeen mukaan. Yleisesti

voidaan todeta, että naulalevyristikkoja suunnittelevat asiaan erikoistuneet suunnittelu-toimistot ja suunnittelijat tähän suunnitelluilla ohjelmistoilla, kun taas teräsristikon suunnittelu on tyypillisesti teräsrakennesuunnittelijan tehtävä, joka tehdään muussakin suunnittelussa käytössä olevilla menetelmillä ja työkaluilla. Näin ollen naulalevyristikot on jätetty tämän tarkastelun ulkopuolelle. Lisäksi tarkastellaan ainoastaan tasoristikoita.

Ristikon kuormankantokapasiteetti määräytyy globaalin geometrian ja topologian kautta perustuen lopulta yksittäisten sauvojen sekä liitosten kestävyys. Lisäksi tarvitaan erityistarkasteluja ristikon globaalin stabiliteetin osalta. Tyypillisessä ristikon rakennemallissa uumasauvat on nivelellisesti liitetty parresauvoihin. Tämä johtaa siihen, että sauvan kapasiteetin määrää vetosauvan normaalivoimakapasiteetti tai puristussauvan nurjahduskapasiteetti. Tyypillisesti ristikon ulkoinen kuormitus kohdistuu joko yhdelle tai molemmille parresauvoille ja liitosten epäkeskisyys saattaa kohdistaa lisämomenttia parresauvoille. Lisäksi paarteiden ja uumasauvojen välisillä liitoksilla on olennainen vaikutus ristikon kapasiteettiin. Vaikka liitoksia voidaan vahvistaa paikallisesti niiden kapasiteetin ylittyessä, johtaa tämä huonoihin ratkaisuihin ristikon kokonaistaloudellisuuden kannalta. Erityistapauksena tukiliitoksissa voi muu kuin paarteen normaalivoimakkestävyys tulla määrääväksi, mutta tällöin sauvoja voidaan vahvistaa paikallisesti.

Koska ristikon topologialla on merkittävä vaikutus yksittäisten sauvojen kestävyys nurjahduspituudesta johtuen, on sillä myös merkittävä vaikutus ristikon massa. Kokenut teräsrakennesuunnittelija pystyy kokemuspohjaisesti näkemään hyvän tai kohtalaisen hyvän ratkaisun topologialle. Menetelmällä ei kuitenkaan tyypillisesti löydetä parasta ratkaisua, minkä lisäksi vaihtoehtojen tutkiminen on hidasta (ja edelleen kallista). Parametrin suunnitteluprosessin avulla voidaan tutkia useita topologiavaihtoehtoja lyhyessä ajassa. Kun topologia parametrisoidaan ja automatisoidaan poikkileikkauksen valinta perustuen sauvassa kullakin vaihtoehdolla vaikuttaviin voimiin, voidaan uusia vaihtoehtoja tarkastella nopeasti muuttamalla lähtöarvoja, eli parametreja.

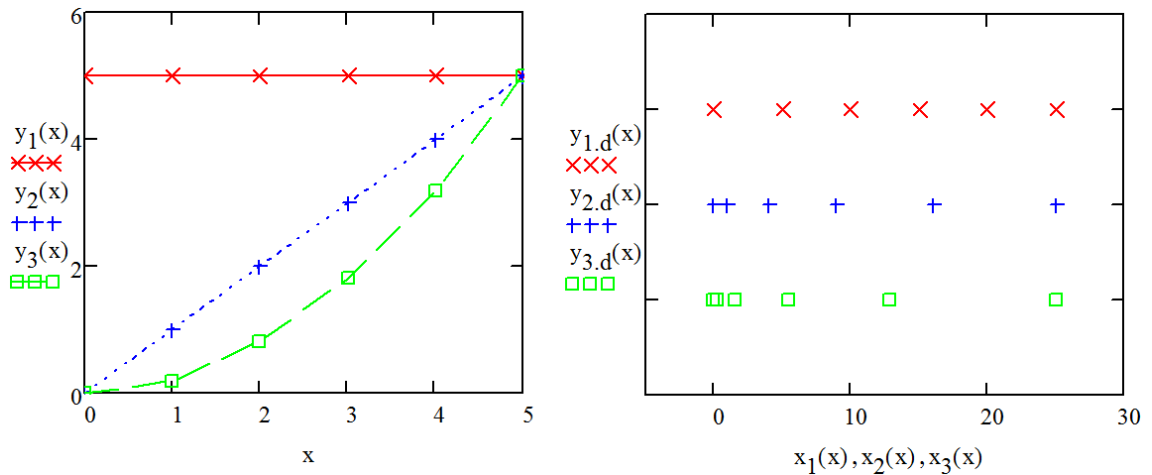


Kuva 5.6. Ristikon parametrinen suunnitteluprosessi.

Ristikon parametrinen suunnittelu etenee prosessina kuvan 5.6 mukaisesti noudattaen pääpiirteissään yleistä parametrin suunnittelun prosessia. Ensin valitaan ristikon parametrit. Parametrit on järkevää valita siten, että ristikon sauvoittelua voidaan muuttaa mah-

dollisimman laajalla skaalalla. Parametreja voivat olla esimerkiksi kattoristikon tapauksessa kattokaltevuus sekä ristikon korkeus. Tällaisten ristikon ulkomuodon geometriaa määräävien parametrien valinnan järjestyminen riippuu siitä, ovatko nämä asiat rakennesuunnittelijan valittavissa. Hyviksi parametreiksi havaittiin Grasshopper-ympäristössä paarteiden ja uumasauvojen keskiviivojen teoreettiset leikkauspisteet. Näiden pisteiden avulla voidaan ristikon topologiaa, kuten sauvojen lukumäärää ja kulmaa hallita täysin. Lisäksi niiden avulla ristikko voidaan linkittää muihin ohjelmistoihin, kuten FEM-ohjelmistoon tai mallinnusohjelmistoon. Käyttämällä esimerkiksi sauvojen kulmia parametreina, tulee leikkauspisteet laskea joka tapauksessa erikseen, mikäli parametrin mallin tuloksia halutaan jatkokehittää muissa ohjelmistoissa.

Parametrien valinnan jälkeen luodaan parametrinen malli, jossa määritellään parametrien vaikutukset ristikkoon ja luodaan ristikon geometria. Kuten edellä on esitetty, parametrien vaikutusten määrittely on suoraa seurausta parametrien valinnasta ja tarkoittaa parametrien lukitsemista määrittämällä sääntöjä. Ristikon yksiselitteiseen määrittämiseen tarvitaan jokin määrä parametreja (joka riippuu parametrien luonteesta). Näin ollen ne parametrit, joita ei ole valittu vapaiksi tulee määrittää parametrin mallin säännöillä. Mikäli parametrit esimerkiksi yhden paarten osalta valitaan siten, että ensimmäinen parametri on leikkauspisteiden lukumäärä, voidaan toinen (tai useampi) parametri valita leikkauspisteiden keskinäiseksi etäisyydeksi. Jokaisen välin keskinäisen etäisyyden valitseminen omaksi parametrikseen voidaan katsoa huonoksi valinnaksi, koska välien lukumäärä ja näin ollen parametrien lukumäärä on muista parametreista riippuvainen. Tämä johtaa mallin toimivuusongelmiin, ellei siihen varauduta erityisillä toimenpiteillä. Näin ollen keskinäisen etäisyyden tulee olla riippuvainen jostain funktiosta eli säännöstä. Toisaalta tämä funktio tai sen jokin vakio voidaan valita parametrikseksi. Kuvassa 5.7 on havainnollistettu parametrin mallin sääntöjä leikkauspisteiden välisen etäisyyden tapauksessa.



Kuva 5.7. Parametrien lukitseminen säännöillä.

Kuvassa 5.7 vasemmalla on esitetty kolme mahdollista sääntöä kuvaamaan ristikon sauvojen leikkauspisteiden välisiä etäisyyksiä. Esitetyt säännöt ovat muotoa (indeksointi vastaa kuvan indeksointia):

$$y_1 = b \quad (15)$$

$$y_2 = kx * b \quad (16)$$

$$y_3 = ax^2 + bx + c, \quad (17)$$

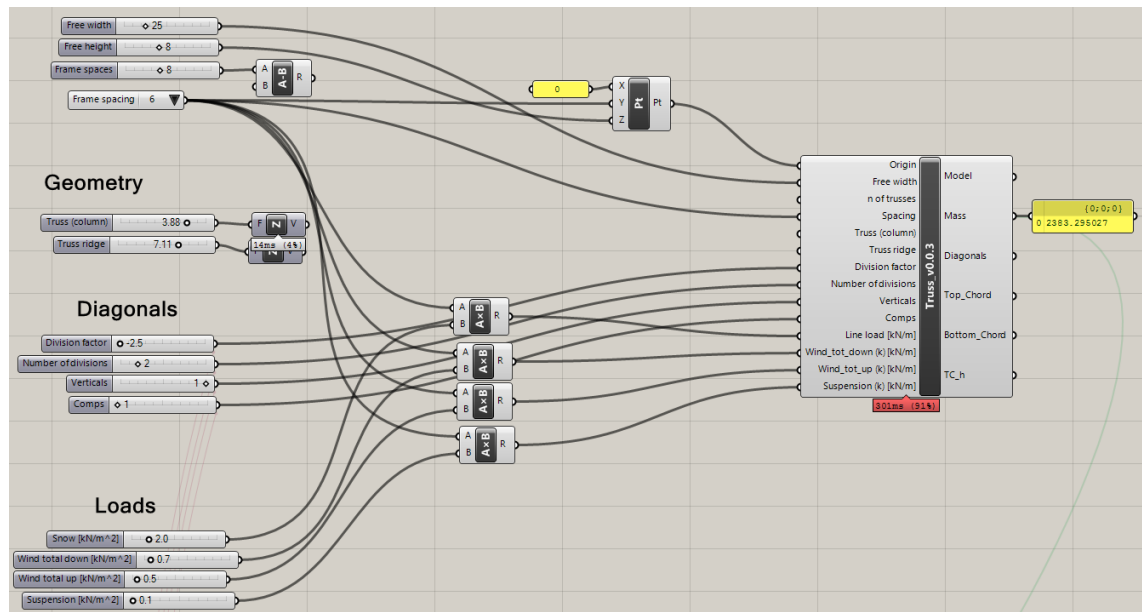
missä y kuvaa vierekkäisten pisteiden välistä etäisyyttä, x on tarkasteltavan pisteen järjestysnumero (diskreetti muuttuja, viiva kuvaa trendiä) ja k , a , b , ja c ovat vakioita. Kuvan oikealla puolella on esitetty pisteiden sijainti x -akselilla (yksikötön etäisyys) vastaavilla funktioilla määriteltynä. Pisteet on esitetty eri korkeuksilla kuvan selkeyttämiseksi, vaikka tarkastellaan 1-ulotteista muuttujaa (etäisyys). Kuvasta havaitaan, että sääntöjen määrittelyllä on merkittävä vaikutus lopputulokseen. Lisäksi havaitaan, että erääksi mallin parametriksi voidaan valita mikä tahansa määrittelyfunktion vakio tai jopa määrittellä itse funktio parametriksi, kuten edellä on esitetty.

Kun ristikon geometria on yksiselitteisesti määriteltä parametrien ja sääntöjen avulla ja ristikkoon vaikuttavat ulkoiset kuormat tunnetaan, voidaan laskea ristikossa vaikuttavat voimasuureet. Käytännössä ne lasketaan mitoitus varten FEM-laskennalla. Voimasuureiden sekä geometriamallin avulla määritettyjen nurjahduspituuksien avulla voidaan mitoitaa sauvojen poikkileikkaukset. Myös poikkileikkausten valinnassa voidaan hyödyntää parametreja ja sääntöjä, koska tyypillisesti ristikon jokaista sauvaa ei mitoiteta erikseen (valita kullekin lokaalia optimia), vaan rajoitutaan uumasauvojen osalta esimerkiksi 3...5 erilaiseen poikkileikkaukseen. Tällöin parametreilla voidaan esimerkiksi valita sallittu erilaisten sauvojen lukumäärä ja säännöillä taas valita, mille sauvoille asetetaan samanlainen poikkileikkaus. Esimerkiksi kattoristikkojen tapauksessa voidaan tyypillisesti hyödyntää tietoa, että kaikki suurimmat kuormitustapaukset aiheuttavat samansuuntaiset normaalivoimat uumasauvoihin (ja paarteisiin). Tällöin sauvoista on yksinkertaisilla

säännöillä helppo valita esimerkiksi eniten puristetut sauvat ja antaa niille riittävän suuren taivutusjäykkyyden omaava poikkileikkaus.

Kun topologia ja sauvojen poikkileikkaukset on valittu, on ristikko perinteisen menetelmän kriteereillä suunniteltu, jos se täyttää kaikki asetetut suunnitteluehdot. Parametrisen mallin avulla on kuitenkin luotu ristikko, joka on ”jatkovasti” suunniteltu riippumatta parametrien arvoista (olettaen että parametrien arvot on rajattu alueelle, jossa ratkaisu on olemassa). Näin ollen parametreja muuttamalla voidaan muuttaa ristikon topologiaa ja poikkileikkauksia siten, että ristikko täyttää aina asetetut suunnitteluehdot. Tämä mahdollistaa ristikon ominaisuuksien tarkastelun muustakin kuin kantavuuden näkökulmasta (koska tämä ehto täytetään automaattisesti). Tällaisia ominaisuuksia ovat esimerkiksi ulkonäkö ja massa, joka on suoraan verrannollinen ristikon hintaan. Parametrinen suunnittelu mahdollistaa siis massan optimoinnin parametreja variomalla, mikä voidaan toteuttaa myös optimointialgoritmin avulla. Tätä käsitellään luvussa 5.6. Voidaan todeta, että ristikkojen parametrilla mallintamisella saavutetaan mahdollisuus muuttaa ristikon geometriaa (ulkomittoja) esimerkiksi luonnossuunnitteluvaiheessa sekä optimoida ristikon ominaisuuksia asetettujen tavoitteiden suhteen.

Kuten muidenkin rakenneosien, myös ristikkojen mallinnus voidaan toteuttaa komponenttien avulla. Edellä mainittuja periaatteita noudattamalla on luotu ristikkokomponentti Grasshopper-ympäristöön. Se on esitetty kuvassa 5.8.

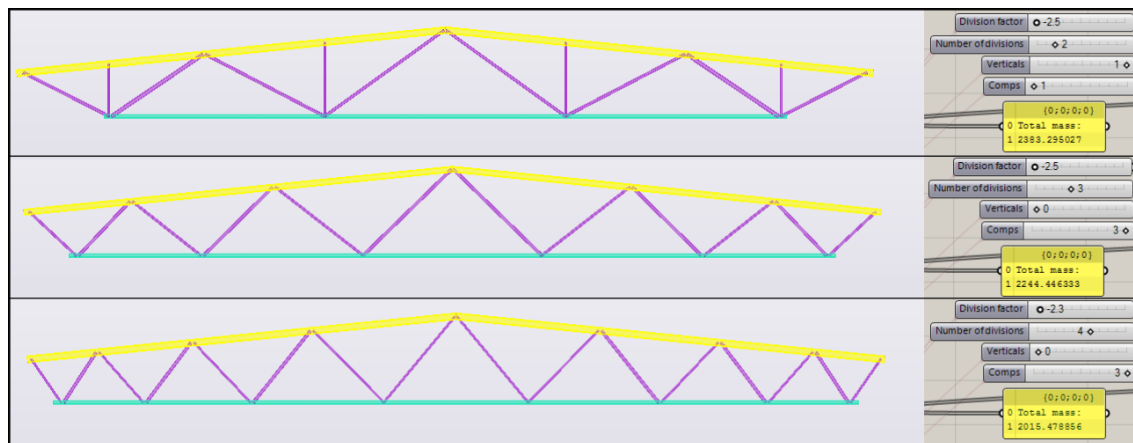


Kuva 5.8. Ristikkokomponentti.

Kuvan 5.8 komponentille annettavia parametreja ovat sijainti, vapaa leveys, ristikoiden lukumäärä ja ristikkoväli, ristikon ulkoisen geometrian parametrit, topologiaparametrit sekä kuormaparametrit. Ulkoisen geometrian parametreille on annettu vakioarvot tyypillisen hallin tapauksessa, jotka voidaan tarvittaessa yliajaa, mikäli suunnittelukohde tällaista vaatii. Kuormaparametreilla määrätään ristikon rasitustila ja topologiaparametreilla

määrätään uumasauvojen sijoittelua. Lopputuloksena saadaan ristikon geometria poikkileikkauksineen sekä tietoa ristikosta, olennaisimpana tietona massa.

Komponentin avulla voidaan nyt luoda esimerkiksi kuvan 5.9 kaltaisia ristikoita.



Kuva 5.9. Ristikkokomponentilla luotuja ristikkoja.

Kuvassa 5.9 on esitetty kolme vertailutapausta, jotka on luotu samalla ristikkokomponentilla käyttäen samoja kuormitusparametreja sekä ulkoisen kuorman parametreja. Ainoastaan topologiaparametreja on muutettu. Ensimmäisenä hyötynä komponentin käytössä havaitaan, että erilaisia ristikkovaihtoehtoja kyetään luomaan ja suunnittelemaan sekunneissa. Näin ollen voidaan toteuttaa kuvan kaltainen vertailu, jossa ristikon topologiaa muuttamalla saadaan erilaisia ristikoita, joiden massat vaihtelevat. Lyhyen vertailun tuloksena löydetään alin ristikko, jonka massa on noin 350 kg kevyempi kuin ylimmän vaihtoehdon. Tämä ratkaisu voidaan todeta myös hitsattavuudeltaan huonoksi vaihtoehdoksi. Edelleen alin ristikko on noin 200 kg kevyempi kuin keskimääräinen vaihtoehto. Lisäksi komponentilla voidaan kopioida ristikko tietomallissa jokaiseen hallin kehäväliin, jolloin tehokkuus kasvaa mallintamisessa, suunnittelussa sekä materiaalin hyödyntämisessä.

5.3.5 Liitokset

Liitosten mallintaminen asettaa suurempia vaatimuksia parametriselle suunnittelulle. Tämä siksi, että liitosten toiminnallinen analyysi suoritetaan perinteisesti monimutkaisilla laskelmilla tai vaihtoehtoisesti FEM-analyysillä ja ne sisältävät edellä esitettyihin verrattain monia parametreja. Lisäksi liitosten geometria monimutkaisempaa, kuin yksittäisistä janoista muodostuvien rakenneosien.

Jos parametrasta mallintamista ajatellaan alustariippumattomasti, yrityksillä on tyypillisesti käytössään liitosten laskentaan soveltuvia parametrisia laskentapohjia. Tällaisissa pohjissa parametreina ovat esimerkiksi voimasuureet, liittyvät poikkileikkaukset, ruuvien määrä sekä ruuvien koko. Tyypillisesti reunaehdoista ovat suunnittelunormiston (esimerkiksi eurokoodi) esittelemät perustapaukset, joita voidaan soveltaa käsinlaskennassa sekä

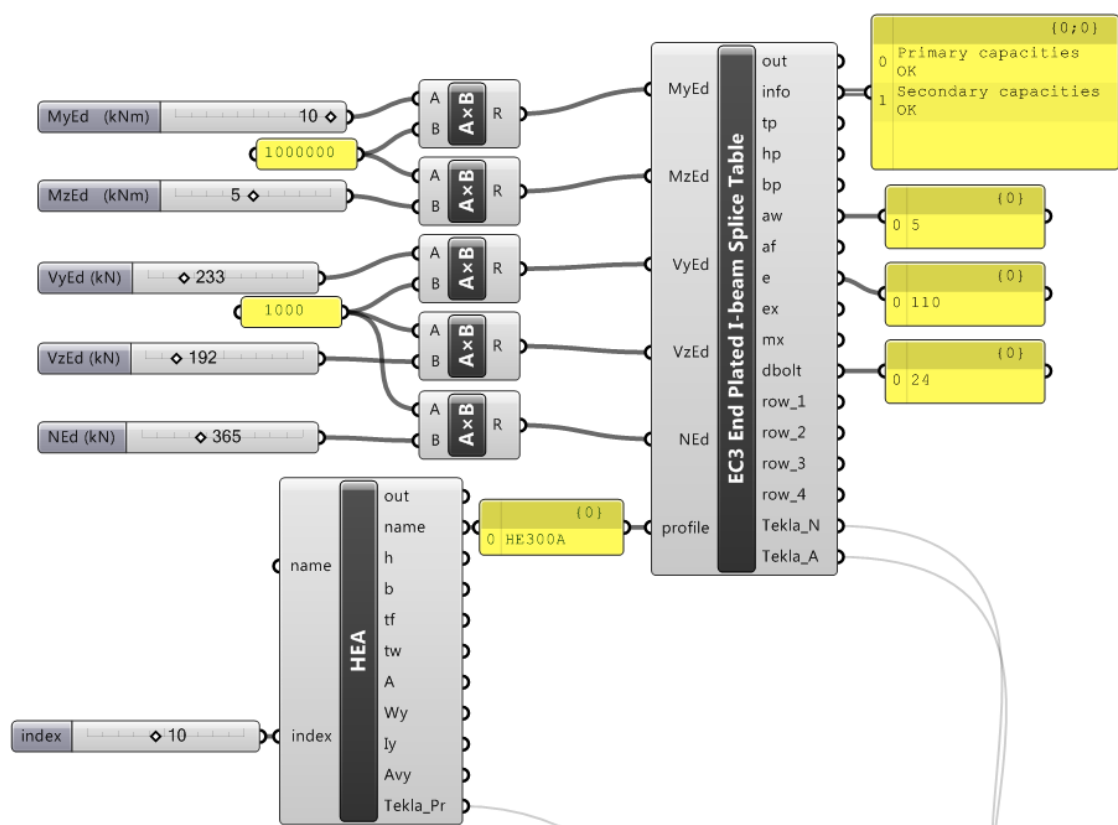
rajoittuminen ainoastaan yhteen liitostyyppiin laskentapohjaa kohti. Rajoittamalla liitostyyppien määrä yhteen yhdessä laskentapohjassa varmistetaan pohjan selkeys, käyttökelpoisuus ja käyttäjäystävällisyys. Ottaen nämä reunaehdot huomioon voidaan todeta, että liitosten suunnittelu sopii tämänkaltaiselle menetelmälle varsin hyvin.

Edellä esitetyn tulkinnan perusteella liitoksia voidaan tai niitä on jopa järkevää suunnitella parametrisesti. Koska visuaalisessa ohjelmoinnissa on mahdollista suorittaa myös eri ohjelmointikielillä toteutettuja skriptejä eli ohjelmointikielellä kirjoitettuja matemaattisia ja yksiselitteisiä ohjeita, voidaan liitoksia näin ollen suunnitella myös parametrisesti myös visuaalisissa ohjelmointiympäristöissä.

Liitosten parametriseen suunnitteluun voidaan esittää kolme erilaista menetelmää:

1. taulukkomenetelmä
2. laskentamenetelmä
3. täysin parametrinen menetelmä.

Taulukkomenetelmä vastaa perinteisen suunnittelumenetelmän taulukkomitoitusta. Parametriseen ympäristöön kirjataan taulukkomaisesti liitoksen kapasiteettiin vaikuttavat tekijät sekä hyväksymisehdot. Parametriselle taulukkokomponentille voidaan näin antaa parametreina liitoksen rasitukset sekä muut vaikuttavat tekijät, kuten esimerkiksi liittyvät poikkileikkaukset. Näiden perusteella komponentti valitsee liitoksen perustuen annettuihin taulukkoarvoihin. Kuvassa 5.10 on esitetty esimerkki tällaisesta liitossuunnittelusta Grasshopper-ympäristössä.



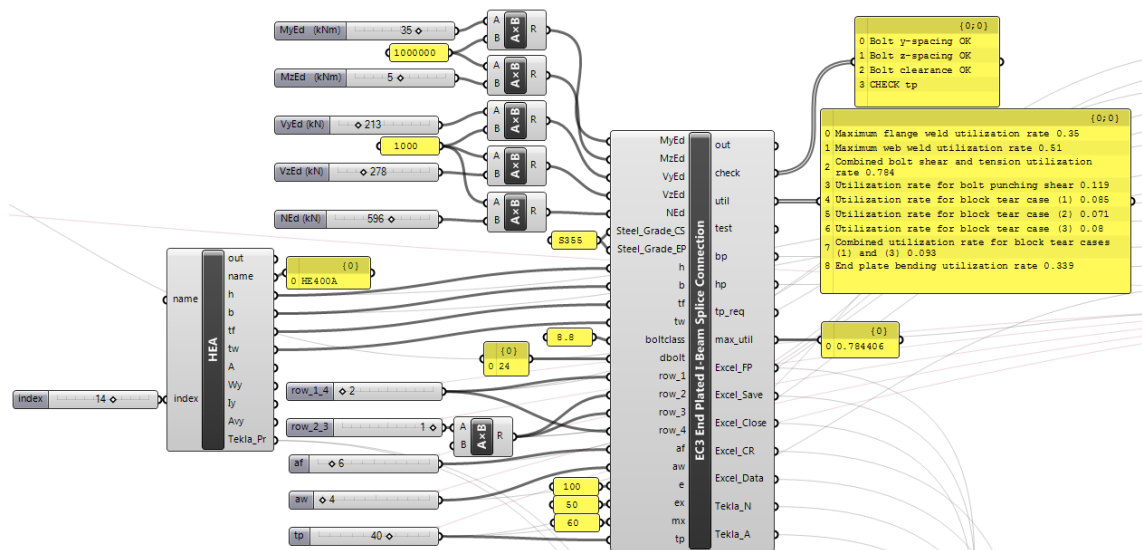
Kuva 5.10. Esimerkki liitoksen parametrisesta taulukkosuunnittelusta.

Kuvassa 5.10 komponentti "HEA" sisältää tiedot erilaisten HEA-poikkileikkausten geometriasta, joita liitossuunnittelussa tarvitaan. Tämän yläpuolella on määrätty liitokseen kohdistuvat rasitukset. Komponentti "EC3 End Plated I-beam Splice Table" sisältää liitostaulukon, jonka perusteella käytettävä liitos valitaan sekä algoritmin, joka valitsee liitoksen. Kuvassa on myös näkyvissä kapasiteettitarkastelun lopputulos, uimahitsin a-mitta sekä tarvittava ruuvien halkaisija. Komponentilla saadaan tietysti myös muut liitoksen geometriatiedot, joita voidaan edelleen hyödyntää parametrisessa mallintamisessa tai linkityksessä muihin ohjelmistoihin. Menetelmän etuna on komponenttien tekemiseen kuluva aika, joka on lyhyt verrattuna muihin menetelmiin. Koska komponentti on suhteellisen yksinkertaista luoda, eikä se välttämättä vaadi ohjelmointiosaamista, on niitä mahdollista luoda myös rakennesuunnittelijan perusosaamisella. Koska varsinaista laskentaa ei suoriteta komponentissa, vaan se on tehty valmiiksi, on suorittamiseen kuluva aika (vaadittu laskentateho) verrattain lyhyt, joten komponentti ei hidasta suuriakaan parametrisia malleja.

Toinen tapa suunnitella liitoksia parametrisesti on laskentamenetelmä. Tämä vastaa käytännössä perinteisessä suunnittelussa käytettäviä laskentapohjia. Laskentapohja voidaan toteuttaa esimerkiksi visuaalisessa ohjelmointiympäristössä tai jollakin ohjelmointikielellä tai linkittää ulkoinen laskentaohjelmisto tai -pohja osaksi algoritmia. Monissa tapauksissa liitosten laskentapohjasta tulee melko monimutkainen, mikäli kaikki liitokseen vaikuttavat tekijät halutaan parametrisoida. Tällöin osoittautuu, että graafinen ympäristö

nykyisessä muodossa kirjoittamishetkellä osoittautuu kömpelöksi tällaisten pohjien toteuttamiseen, johtuen erilaisten perinteisessä ohjelmoinnissa käytettyjen ominaisuuksien, kuten silmukoiden puuttumiseen. On kuitenkin muistettava, että tietotekniikka, kuten ohjelmointikin on nopeasti kehittyvä ala, joten tämä saattaa muuttua nopeastikin.

Laskentamenetelmässä komponentille annetaan vastaavat parametrit kuin vastaava laskentapohjakin tarvitsisi eli käytännössä kaikki liitoksen kestävyyyteen vaikuttavat tekijät. Näiden perusteella komponentti suorittaa laskennan, jonka tuloksena saadaan esimerkiksi liitoksen eri osien käyttöasteet tai kokonaiskäyttöaste, parametrien perusteella mitoitettua geometriaa sekä tietoa liitoksen sovellettavuudesta, kuten tieto ruuvien minimietäisyyksien täyttymisestä. Kuvassa 5.11 on esitetty esimerkki tällaisesta liitossuunnittelusta Grasshopper-ympäristössä.

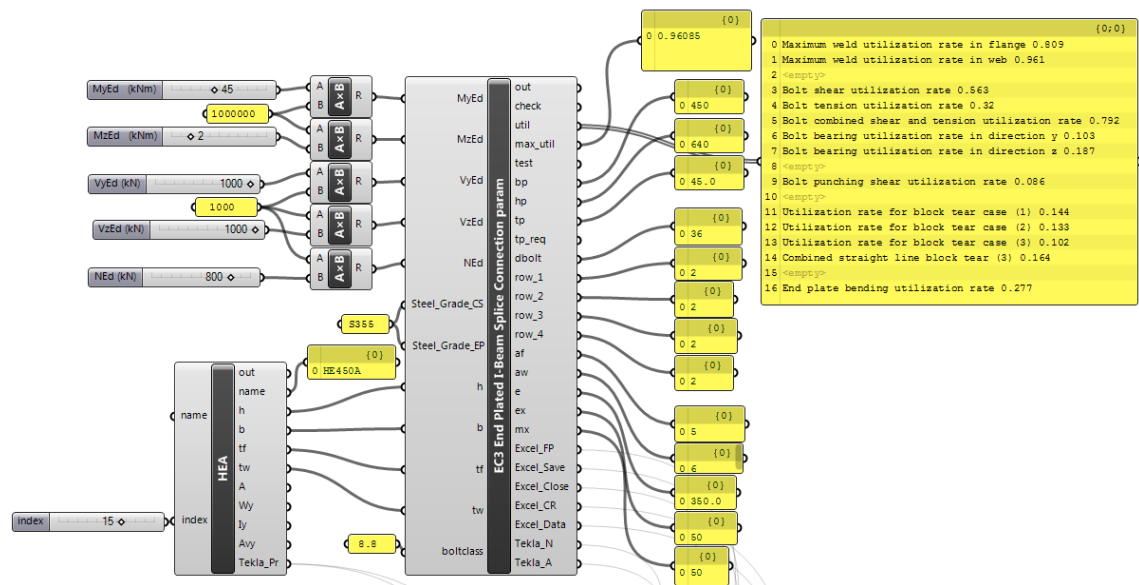


Kuva 5.11. Esimerkki laskentamenetelmään perustuvasta liitossuunnittelusta.

Kuvassa 5.11 liitoskomponentille (*“EC3 End Plated I-Beam Splice Connection”*) annetaan taulukkomenetelmää vastaavalla tavalla poikkileikkauksen tiedot sekä rasitustiedot. Muita tarvittavia parametreja ovat tässä tapauksessa terästen lujuusluokat, ruuvien lujuusluokat, ruuvien halkaisija, ruuvien määrä rivissä sekä hitsien koot. Tuloksena saadaan tieto käyttöasteista sekä komponentissa suunniteltua geometriatietoa. Menetelmässä on syytä kiinnittää huomiota tarvittavien parametrien suureen määrään. Komponentti jättää itse suunnittelun käyttäjän tai käyttäjän luomien algoritmien toteutettavaksi ja suorittaa ainoastaan kapasiteettitarkistuksen. Menetelmän etuna on taulukkomenetelmää laajempi sovellettavuus. Suuri auki olevien parametrien määrä mahdollistaa myös liitoksen optimoinnin esimerkiksi evoluutiomenetelmillä. Lisäksi komponentin suorittamiseen kuluva laskenta-aika on suhteellisen lyhyt, ja komponenttia pystytään hyödyntämään parametrisen mallin osana. Huonoja puolia on komponentin tekemiseen kuluva aika, joka on merkittävästi suurempi, kuin taulukkocomponentin luomiseen kuluva aika. Koska komponentti ei suunnittele liitosta, tulee käyttäjän toteuttaa käsin liitoksen suunnittelu.

Edellä esitetystä menetelmästä liittostyyppien määrä rajoitettiin yhteen komponenttia kohti, kuten tyypillisesti toteutetaan laskentapohjia. On kuitenkin mahdollista toteuttaa komponentti siten, että yhtenä parametrina on juuri käytettävä liitos, ja kuhunkin liitokseen liittyvä mitoitusosa toimii komponentin sisällä toisistaan riippumattomana osana. Valitun liittostyyppin mukaan komponentti suorittaa nimenomaan tähän liitokseen liittyvän laskennan ja antaa tulokset sen perusteella. Tällaisella komponentilla liittostyyppin valinta olisi mahdollista suorittaa algoritmisesti ennen komponentin suorittamista perustuen liitosparametreihin, esimerkiksi rasiituksiin.

Kolmas suunnittelutapa on täysin parametrinen suunnittelumenetelmä. Menetelmä on kehitettyempi versio toisena esitellystä laskentamenetelmästä. Menetelmässä laskentamenetelmän komponenttia on kehitetty siten, että se suunnittelee liitoksen rasiitusten perusteella. Tämä voidaan toteuttaa esimerkiksi antamalla komponentin laskentaosuudelle parametrien alkuarvot ja tämän laskennan tuloksena komponentti etsii iteratiivisesti parametreille arvot, joilla vaaditut ehdot, kuten käyttöasteen jääminen alle 100 %, toteutuvat. Kuvassa 5.12 on esitetty esimerkki tämän menetelmän soveltamisesta Grasshopper-ympäristössä.



Kuva 5.12. Esimerkki täysin parametrisesta liittossuunnittelukomponentista.

Kuvassa 5.12 suunnittelukomponentille ("EC3 End Plated I-Beam Splice Connection param") annetaan tiedot liittyvistä poikkileikkauksista sekä vaikuttavista rasituksista, kuten edeltävissäkin esimerkeissä. Tässä esimerkissä parametreista on jätetty auki myös käytettävät materiaalit sillä oletuksella, että käyttäjä haluaa määrätä nämä itse. Olennaista on kuitenkin huomata tulopuolen (komponentin vasen puoli) syötteiden määrä verrattuna kuvassa 5.11 esitettyyn. Komponentille ei siis anneta syötteenä liitososien tietoja kuten laskentamenetelmässä, vaan komponentti suunnittelee ne automaattisesti. Vastaavasti tulopuolelta (komponentin oikea puoli) saadaan laskentamenetelmää vastaavat tiedot suunniteltuna siten, että vaaditut ehdot täyttyvät. Menetelmän etuna saavutetaan täysin

automatoitu liittosuunnittelu. Menetelmän suurin haitta on komponentin luomisen vaikeus. Tällä hetkellä vastaavaa komponenttia ei ole mahdollista luoda visuaalisen ympäristön perustyökaluilla, vaan tarvitaan osaamista jostakin ohjelmointikielestä. Visuaalisille ohjelmistoille on olemassa kolmannen osapuolen luomia lisätyökaluja, joilla tämä saattaa olla mahdollista, mutta niitä ei tässä työssä tarkasteltu. Komponentin luominen vaatii huomattavasti enemmän aikaa, kuin kahden edellä esitetyn tapauksen komponenttien luominen. Lisäksi komponenttien luomiseen tarvitaan vähintään perusteiden hallitsemista ohjelmoinnista, jotta komponentista voidaan luoda riittävän tehokas laskentatoteutus näkökulmasta.

Suunnittelumenetelmän valinta sekä liitosten mallintaminen ja sovellettavuus parametrissa suunnittelussa riippuvat parametrin mallin tavoitteesta. Tyypillisesti parametrissa mallia halutaan hyödyntää luonnossuunnitteluvaiheessa. Tällöin yksittäisen liitoksen geometrinen tarkkuus ei ole vielä kovin olennaista, jolloin taulukkomenetelmä soveltuu tähän hyvin yksinkertaisuutensa ja nopeutensa vuoksi. Lisäksi taulukkomenetelmällä tarvittava komponentti on nopeasti luotavissa, mikäli sellaista ei ole. Mikäli mallia halutaan käyttää eri vaihtoehtojen kustannusvertailuun, on taulukkomenetelmää mahdollista edelleen yksinkertaistaa esimerkiksi taulukoimalla liitosten kustannustietoja ilman geometriaa. Kehittyneempien menetelmien käyttö mahdollistaa osaltaan parametrin mallin hyödyntämisen aina toteutussuunnitteluvaiheeseen asti. Ongelmana on, että mikäli tarvittavat komponentit puuttuvat, on niiden luominen aikaa ja osaamista vaativa työ.

5.3.6 Muut rakenneosat

Parametrin suunnitteluprosessin soveltuvuutta muihin rakenneosiin ei tämän työn yhteydessä tutkittu esiselvitystä enempää. Olennaisina soveltuvuutta määrittävinä asioina voidaan pitää luvussa 4.3.1 esitettyjä taloudellisuusnäkökulmia. Lisäksi on syytä tarkastella käytössä olevien työkalujen soveltuvuutta parametriseen suunnitteluun.

Lähtökohtaisesti parametriseen suunnittelusta on saavutettavissa hyötyä rakenteiden tietomallintamisessa rakenneosasta riippumatta geometrialtaan haastavissa rakenneosissa. Rakenneosan haastava geometria voidaan mallintaa tähän paremmin soveltuvassa ympäristössä, kuten Rhinoceros/Grasshopper ympäristössä, ja siirtää varsinaisten rakenne-suunnitelmien tekemiseen käytettävään ympäristöön, kuten Tekla Structures -ohjelmaan. Hyöty ja siitä johtuva soveltuvuus parametriseen mallintamiseen perustuu siis tehokkaammasta mallinnustavasta seuraavaan ajan säästöön.

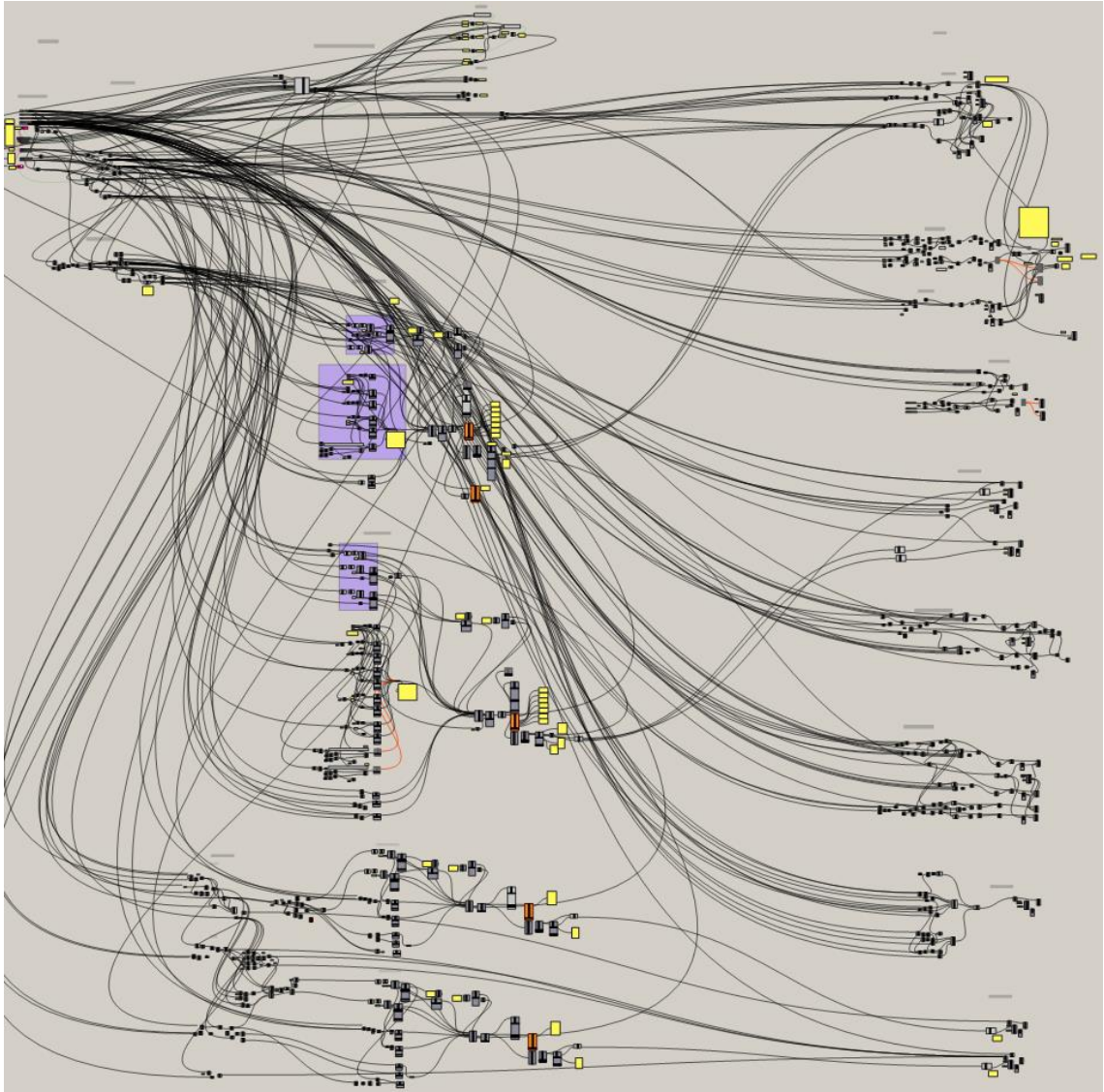
Rakenteelliseen kestävyystarkasteluun liittyvään suunnitteluun asettavat rajoitteita erityisesti ohjelmistojen väliseen tiedonsiirtoon liittyvät ongelmat. Tähän törmättiin tämän tutkimuksen aiheenrajausvaiheessa erityisesti betonirakenteiden osalta. Vaikka teräsbetoninen rakenneosa voidaan suunnitella parametrisesti aivan vastaavasti kuin muustakin materiaalista tehty rakenneosa, ei suunnittelun raudoituksen siirtäminen mallinnusohjelmasta suunnitelmien tuottamiseen käytettyyn ohjelmistoon ole niin yksinkertaista kuin

homogeenisen rakennusmateriaalin tapauksessa. Tämä johtaa edelleen käsityöhön suunnitelmien luonnissa ja niiden muuttuessa, mikä lähtökohtaisesti vähentää parametrinen suunnittelun etuja suhteessa perinteisiin suunnittelumenetelmiin.

Yhteenvetona voidaan todeta, että rakenneosan soveltuvuuden parametriseen suunnitteluun määrää saavutetaanko sillä hyötyä verrattuna perinteiseen suunnittelumenetelmään. Tarkasteltujen rakenneosien perusteella voidaan todeta, että todennäköisin etu parametrinen suunnittelun hyödyntämisestä on rakennesuunnittelijan työmäärän pieneminen. Parametrinen suunnittelu on käsitteenä niin laaja, että mikäli hyötyjä voidaan tunnistaa ja voidaan esittää keinot niiden saavuttamiseksi, soveltuu parametrinen suunnittelu tällaisen rakenteen suunnitteluun.

5.4 Parametrinen malli

Edellä esitettyjä yksittäisten rakenneosien parametrisia komponentteja hyödyntämällä soveltuvilta osin voidaan luoda parametrinen malli kokonaisesta rakennuksesta. Kuten luvussa 5.3.3 on esitetty, pilari ei sovellu erityisen hyvin komponenttipohjaisesti suunniteltavaksi, vaan se on järkevä toteuttaa osana rakennejärjestelmää. Lisäksi mallin teräsrungossa esiintyy palkkeja ainoastaan päädyssä, ja ne toimivat rakenteellisesti yhdessä päädyn jäykistysristikon sekä päätypilareiden kanssa. Tällöin on järkevää mallintaa koko pääty yhdessä palkin kanssa lujuuslaskenta varten, koska palkkikomponentilla saataisiin ainoastaan alustavia tuloksia. Näin ollen mallissa on päädytty hyödyntämään esitetyistä komponenteista ainoastaan ristikkokomponenttia. Kuvassa 4.13 on esitetty yksinkertaisen teräsrakenteisen yksilaivaisen hallin parametrinen malli.

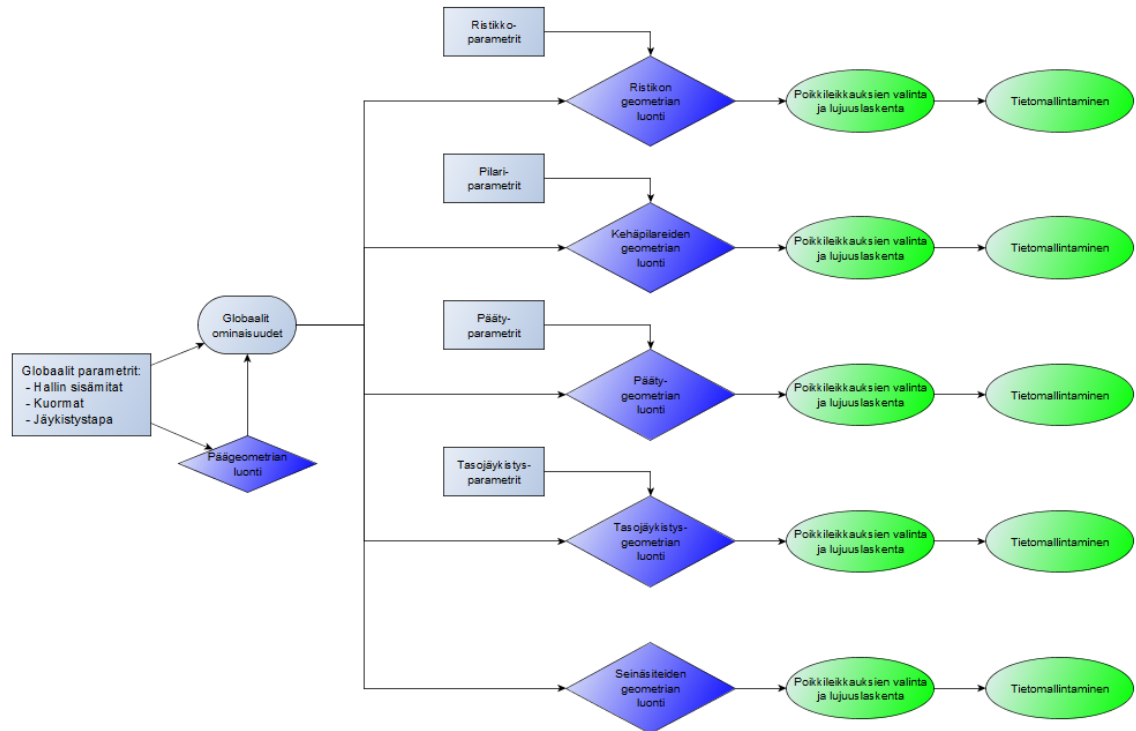


Kuva 5.13. Esimerkki parametrisesta mallista Grasshopper-ympäristössä.

Hallin parametrinen malli koostuu parametrien, geometrian luonnin, kantavien rakenteiden lujuuslaskennan sekä suunnitellun geometrian siirtämisen muodostavista osista. Parametrit on tässä esityksessä valittu kuvaamaan hallin geometrisia ulkomittoja ja vapaita mittoja, kehäjakoa, kuormia sekä yksittäisten rakenneosien ominaisuuksia, kuten ristikon topologiaa. Geometrian luonti toteutetaan hallia varten luodulla algoritmilla, jota säätelevät geometriaparametreilla. Tämä vaikuttaa hallin vapaisiin mittoihin ja ulkomuotoon, kehäjakoon sekä jäykistävien sauvojen geometriaan. Ristikkokomponentilla luodaan ristikon tarvitsema lisägeometria eli ristikon sauvat.

Rakenteellinen lujuuslaskenta toteutetaan kehän kantavan ristikon osalta ristikkokomponentilla, pääpilarit lasketaan niitä varten toteutetulla laskenta-algoritmilla ja hallin päädyt sekä jäykistys lasketaan omina kokonaisuuksinaan. Geometrian siirto mallinnusohjelmaan on toteutettu ristikon osalta komponentilla ja muiden rakenteiden osalta erikseen.

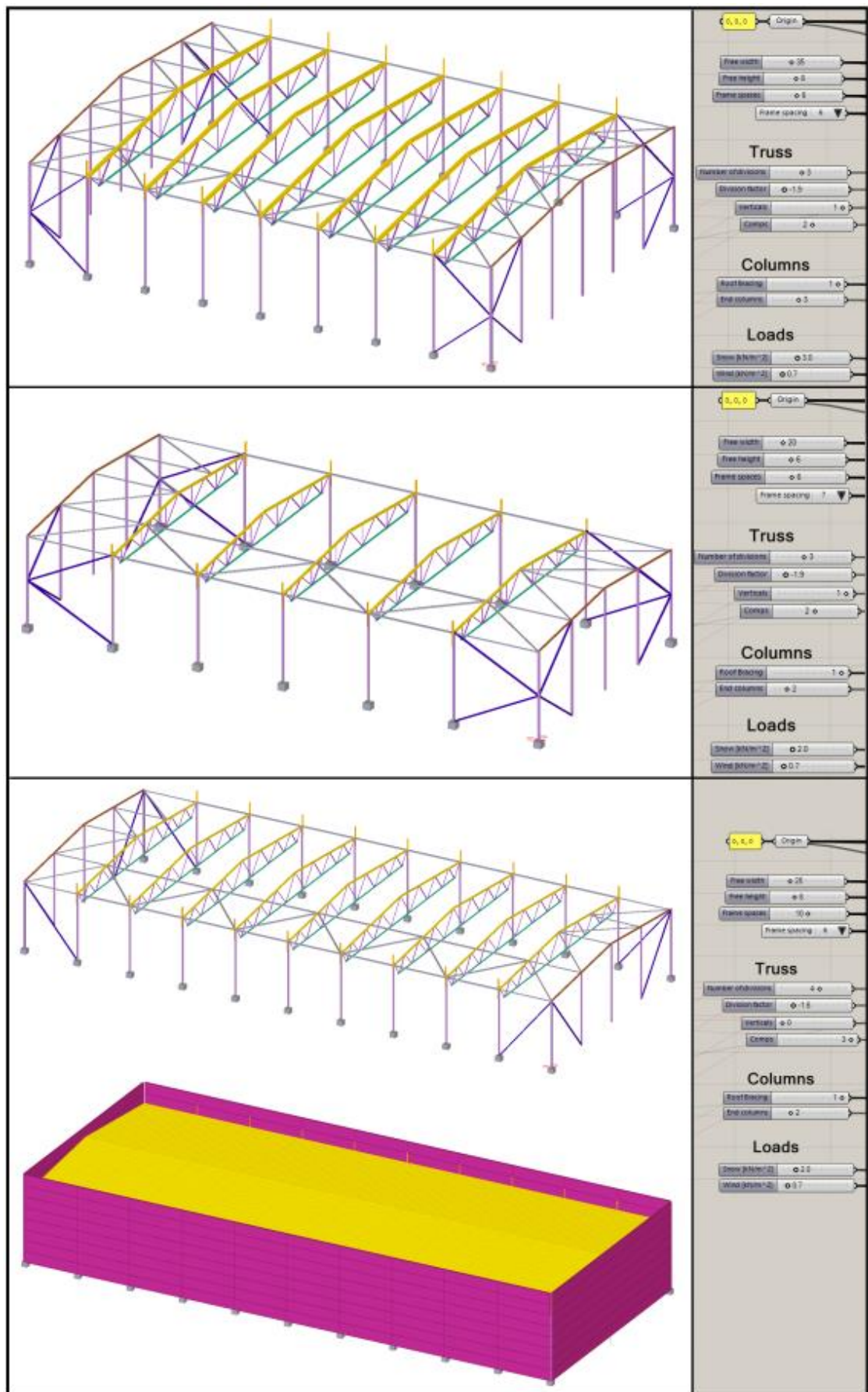
toteutetulla siirtoalgoritmilla. Kuvassa 5.14 on esitetty algoritmin toimintatapa pääpiirteissään.



Kuva 5.14. Hallialgoritmin toimintaperiaate.

Teräshallin algoritmisen mallin tärkeimpiä ominaisuuksia ovat koko hallin massan ja vaippapinta-alojen lasku, joita voidaan hyödyntää luonnossuunnitteluvaiheessa erilaisten vaihtoehtojen vertailuun kustannusten osalta. Mallin avulla voidaan muutamissa sekunneissa vertailla jännemitoiltaan ja pituuksiltaan erilaisten hallien teräsrunkojen massoja ja pinta-materiaalien määriä päästen näin nopeasti käsiksi tilaajan valintojen aiheuttamiin kustannuksiin tai säästöihin.

Parametrinen mallin avulla pystytään visualisoimaan kohde kolmiulotteisena mallina siirtämällä valmiiksi suunnitellut rakenteet suoraan mallinnus ja suunnitteluohjelmistoon, kuten Tekla Structuresiin. Kuvassa 5.15 on esitetty kuvakaappauksia algoritmilla toteutetuista halliratkaisuista.



Kuva 5.15. Parametrisella mallilla tuotettuja hallin tietomalleja.

Kuvassa 5.15 esitettyjen erilaisten vaihtoehtojen luomiseen kuluu vain muutamia sekunteja ulkoisen geometrian parametrien osalta. Näin ollen pystytään nopeasti visualisoimaan ja esittelemään tilaajalle tai muulle taholle mahdollisuuksia ja erilaisten valintojen seurauksia kustannusvaikutusten lisäksi. Alimmassa kuvassa on esitetty viitteellisesti, että algoritmilla voidaan luoda myös pintamateriaaleja, esimerkiksi katon kantavat pöimulevyt. Lisäksi voidaan tutkia rakenneratkaisujen järkevyyttä ja muuttaa tai optimoida niitä rakenneosakohtaisilla parametreilla. Näin nopeasti luodusta alustavasta mallista voidaan tehokkaasti kehittää rakenneratkaisuiltaan yksityiskohtaisempi malli.

Esitetyissä algoritmissa on jonkin verran puutteita. Olennaisimpia puutteita ovat yksiläivaisuus sekä rakennejärjestelmien määrä. Algoritmilla voidaan tehdä ainoastaan yksiläivaisia halleja joko mastopilareilla tai tasojäykistettynä. Algoritmin koekäytössä havaittiin, että jäykistysristikkoihin vaikuttavien parametrien määrä tai riippuvuussuhteet ovat puutteellisia johtaen välillä huonoihin tai jopa toimimattomiin ratkaisuihin jäykistävien rakenteiden osalta. Lisäksi puutteena voidaan mainita tietomallinnus, joka tehdään algoritmissa pääasiassa sauvojen keskiviivojen suhteen teoreettisilla pituuksilla. Tämä johtaa siihen, että malli vaatii käsinmuokkausta rakentamista varten tuotettavien suunnitelmien tuottamiseksi. Havaittiin myös, että algoritmin tuotokset vaativat aina rakennesuunnittelijan kriittisen tarkastelun, jotta virheet, huonot ratkaisut ja luodun algoritmin ohjelmointivirheet havaitaan. Esitetyt puutteet algoritmissa ovat pääasiassa tietoinen valinta, eikä niiden toteuttaminen sisältynyt tutkimuksen tavoitteisiin.

5.5 Yhteistyö muiden suunnittelualojen kanssa

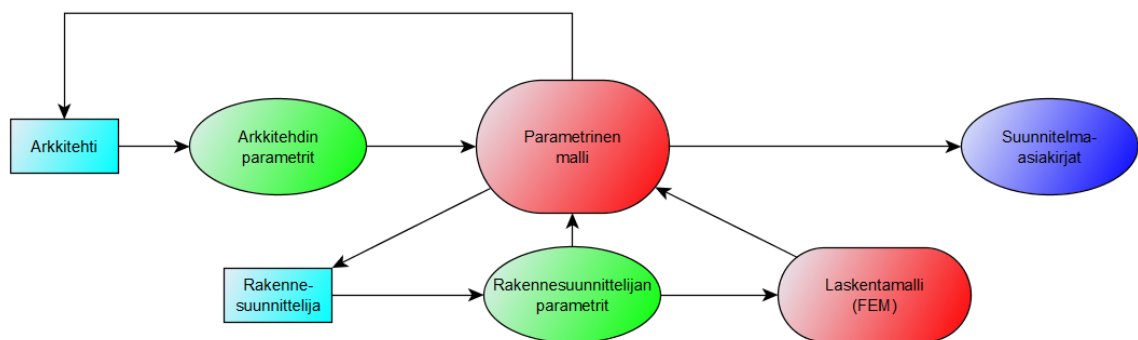
5.5.1 Käytännön yhteistyö

Tyypillisimmillään yhteistyö muiden suunnittelualojen kanssa on yhteistyötä arkkitehdin kanssa. Tästä esimerkkinä on luvussa 5.5.2 esitelty case-tapaus arkkitehdin ja rakennesuunnittelijan yhteistyössä toteuttamasta parametrisesti suunnitellusta kohteesta. Tässä työssä käsitellään käytännön näkökulmasta ainoastaan yhteistyötä arkkitehdin kanssa.

Yhteistyössä arkkitehdin kanssa on kaksi periaatteellisesti erilaista toimintamallia. Ensimmäisessä toimintamallissa yhteistyötä tehdään hyödyntäen samaa parametrista mallia. Luvun 5.5.2 esimerkki on toteutettu tällä mallilla. Molemmat suunnittelualat siis rakentavat reunaehdot ja riippuvuussuhteet samaan malliin. Käytännössä työ aloitetaan arkkitehdin muodostamalla parametrisella geometrialla. Yhteistyössä tähän lisätään rakennesuunnittelijan määrittelemiä reunaehtoja ja riippuvuuksia rakenteellisen toiminnan varmistamiseksi. Yhteistyön tuloksena saadaan malli, joka sovittuja geometriaparametreja varioimalla löytää arkkitehtonisen muotonsa ottaen samalla huomioon rakennetekniset asiat määrätyssä laajuudessa.

Laajuus, jossa rakenteelliset asiat voidaan ottaa huomioon, riippuvat ennen muuta geometrian yksinkertaisuudesta sekä käytettävissä olevista ohjelmistoista. Yksinkertaisen,

käsinlaskettavan geometrian osalta rakennesuunnittelija pystyy halutessaan määrittämään mallin riippuvuussuhteet käsinlaskukaavoin, jolloin malli on aina rakenneteknisesti toimiva. Käytännössä kuitenkin parametrisesti mallinnettava geometria on niin haastava, ettei käsinlasku onnistu. Tällöin parametrisointi voidaan toteuttaa esimerkiksi esisuunnitteluun tarkoitetuilla, parametriseen malliin liitettävillä FEM-ohjelmistoilla tai tunnetuilla nyrkkisäännöillä. Tällaisia nyrkkisääntöjä ovat esimerkiksi tunnetut jännevälin ja rakenekorkeuden suhteet, joilla rakenteen toteuttamisen tiedetään pääsääntöisesti onnistuvan. Molemmissa tapauksissa rakenteen toimivuuden varmistaminen vaatii parametrisesta mallista erillisen laskentakierroksen. Tätä varten rakennesuunnittelija voi edelleen rakentaa parametrisen mallin avulla muuttaa geometriatiedon muotoon, joka on siirrettävissä valittuun laskentaohjelmistoon. Yhteistyömallin prosessikaavio on esitetty kuvassa 5.16.



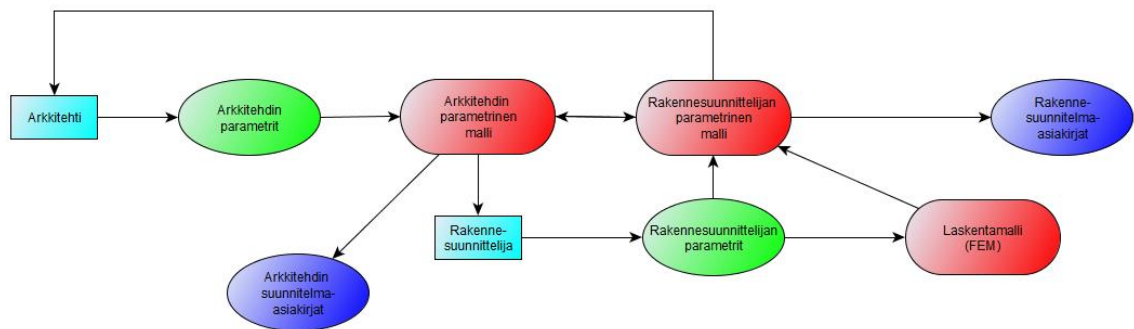
Kuva 5.16. Yhteistyöprosessi arkkitehdin ja rakennesuunnittelijan välillä samassa parametrisessa mallissa.

Kuvan 5.16 prosessikaaviossa projekti alkaa arkkitehdin määrittelemästä parametrisestä mallista. Arkkitehdin mallin avulla rakennesuunnittelija määrittelee omat parametritsa ja liittää ne arkkitehdin malliin. Tarvittaessa voidaan luoda erillinen laskentamalli, joka saa lähtötietonsa rakennesuunnittelijan parametreilta ja välittää laskentatulokset yhteiseen parametriseen malliin. Nyt molempien osapuolet toimivat samassa mallissa ja voivat tarvittaessa tehdä muutoksia siten, että ne vaikuttavat myös toisen suunnittelualan määrittelemien sääntöjen avulla malliin. Molemmat osapuolet tuottavat tarvittavat suunnitelmansa samasta mallista.

Yhteistyömallin hyvänä puolenä voidaan pitää geometriatiedonsiirron helppoutta. Kun molemmat osapuolet toimivat samassa mallissa, on myös alkuperäinen geometria sekä siihen mahdollisesti tehdyt muutokset ovat heti kaikkien käytössä. Toinen hyöty on, että arkkitehti näkee yhteistyössä toteutetusta mallista heti ratkaisujensa rakenteelliset vaikutukset. Tästä on hyötyä erityisesti sellaisissa kohteissa, joissa arkkitehtonisilla ratkaisuilla on poikkeuksellisen suuri rakennetekninen merkitys, kuten siltarakenteissa. Tällöin arkkitehtonisten ratkaisujen vaikutusta rakenteisiin voi olla vaikeampi ennustaa. Toteutustavan haasteena voidaan pitää itse mallinnustyön toteuttamista. Tällä hetkellä ei ole olemassa, joissa useampi käyttäjä voisi työstiä ja käyttää samaa parametrista mallia yhtäaikaaisesti. Näin ollen työ on toteutettava esimerkiksi sovituissa vuoroissa tai muulla tavoin.

työjärjestyksestä sopien. Tätä voidaan pitää myös tietoteknisenä haasteena: voidaanko toteuttaa ohjelmisto, jossa reaaliaikainen yhteistyö on mahdollista?

Toinen yhteistyömalli muistuttaa perinteistä rakennesuunnittelijan ja arkkitehdin yhteistyötä. Yhteistyömallissa kumpikin osapuoli, niin rakennesuunnittelija kuin arkkitehti tekevät työnsä omissa malleissaan ja omilla ohjelmistoillaan. Tiedonvälityksen rajapinnaksi valitaan arkkitehdin tuottama geometria, joka tuotetaan siinä muodossa, että se on suoraan hyödynnettävissä rakennesuunnittelijan mallissa. Kun geometria tuotetaan aina samassa muodossa, muutokset päivittyvät rakennesuunnittelijan parametriseen malliin lähtötietogeometrian päivittämällä. Tämän yhteistyömallin prosessikaavio on esitetty kuvassa 5.17.



Kuva 5.17. Yhteistyöprosessi arkkitehdin ja rakennesuunnittelijan välillä, kun kummatkin osapuolet toimivat omissa malleissaan.

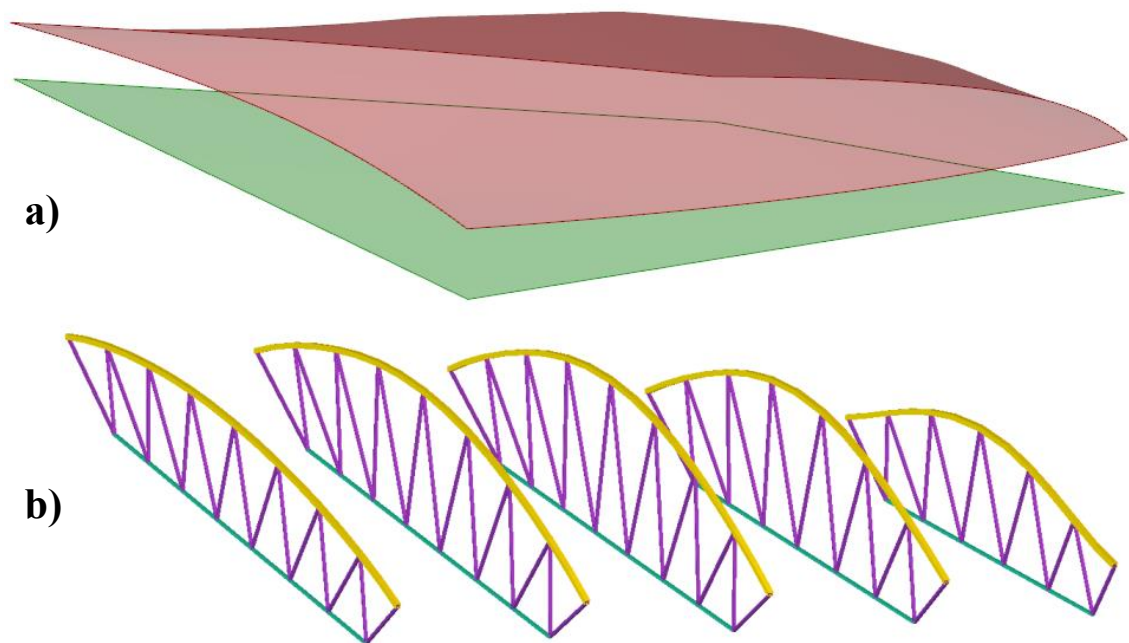
Kuvan 5.17 prosessikaaviossa projekti alkaa jälleen arkkitehdin luomalla parametrisella mallilla. Rakennesuunnittelija luo oman parametrisen mallinsa ja tarvittaessa myös laskentamallin arkkitehdin mallin pohjalta. Molempien parametristen mallien välille sovitetaan tiedonsiirtomekanismi, jonka avulla tieto välittyy suunnittelijalta toiselle. Molemmat tuottavat tarvittavat suunnitelmat omien malliensa avulla.

Yhteistyömallin etuna on totuttu toimintatapa. Molemmat osapuolet voivat tehdä työnsä pääasiallisesti toisistaan riippumatta, kunhan suuria muutoksia alustavaan geometriaan ei tehdä. Lisäksi sovellettavat toimintatavat, menetelmät ja käyttökelpoiset ohjelmistot ovat pääsääntöisesti valmiiksi olemassa. Menetelmän haasteena on rajapinnan toteuttaminen. Jotta rakennesuunnittelijan malli olisi yhteensopiva, tulee reunaehdot rajapinnalle ja käytettäville ohjelmistoille sopia ennen suunnittelun aloittamista. Tämä osaltaan rajoittaa toimintavan käyttökelpoisuutta, mikäli yhteistyömallin luominen ja sen reunaehdoista sopiminen aiheuttaa kohtuuttoman työ määrän suhteessa suunniteltavaan rakenteeseen. Rajapinnan toteuttamiseen on olemassa erilaisia pilvipohjaisia ratkaisuja, kuten Flux ja Konstru, joissa parametrisella mallilla tuotettu tieto ladataan pilveen muiden osapuolten hyödynnettäväksi. Vastaavasti parametrisessa mallissa voidaan hyödyntää muiden osapuolten tuottamaa tietoa. Toimintatapaa voidaan kuitenkin pitää vanhanaikaisena verrattuna ensin esiteltyyn tapaan. Suunnittelun totuttuja toimintatapoja ei tässä pyritä uudista-

maan, ainoastaan suunnitteluun käytettävää menetelmää vaihdetaan. Näin ollen ei ole saatavissa tiiviimmän yhteistyön kautta syntyviä etuja (toisaalta ei myöskään mahdollisia haittoja).

Seuraavassa on esitetty esimerkki toimintatavasta, jossa arkkitehdin ja rakennesuunnittelijan välille on sovittu yhteinen rajapinta. Arkkitehti on määritellyt kupolinmuotoisen, kaksoiskaarevan kattorakenteen yläpinnan ja alapinnan. Pinnat on määritelty yhteiseksi rajapinnaksi ja arkkitehti on sitoutunut toimittamaan ne rakennesuunnittelijan parametriin malliin soveltuvassa muodossa. Rakennesuunnittelija on päättänyt käyttämään teräsrakenteisia ristikkoja ja mallintamaan ne parametrisesti niiden monimutkaisen geometrian vuoksi.

Esimerkissä arkkitehdin antamat pinnat on jaettu pystysuunnan (z-akseli) suuntaisilla tasoväleihin. Näihin tasoihin on sen jälkeen luotu algoritmi teräsristikkoja varten siten, että ristikon paarteet noudattavat arkkitehdin määräämää geometriaa. Lopputuloksena saadaan kuvan 5.18. mukainen rakenne.



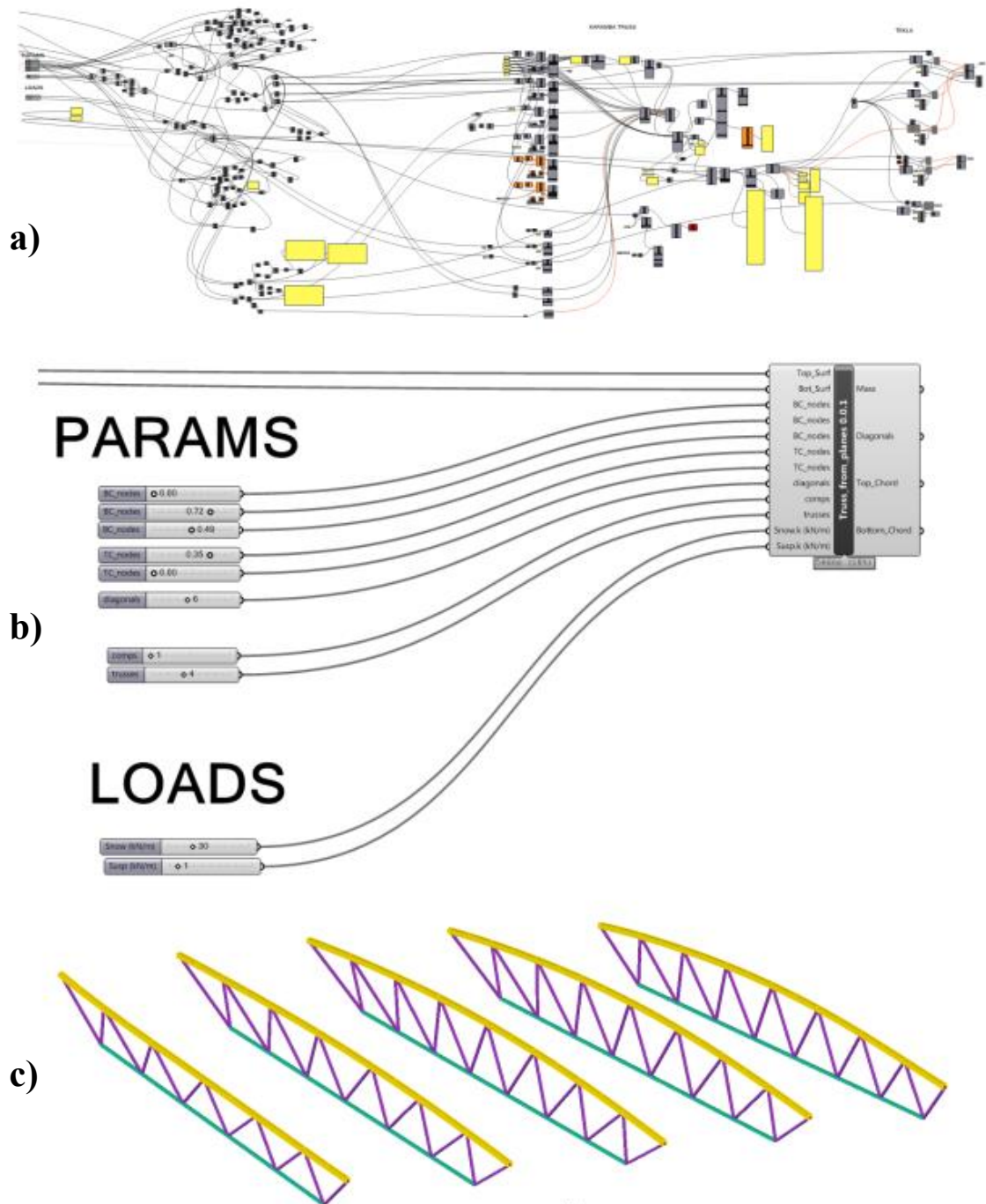
Kuva 5.18. Annetun geometrian perusteella määritetty ristikkorakenne.

Kuvassa a) on esitetty annettu geometria; punaisella yläpinta ja vihreällä alapinta. Kuvassa b) on esitetty geometrian perusteella luodut ristikot, joiden paarteet noudattavat annettuja pintoja. Algoritmi on luotu siten, että se esisuunnittelee ristikon ja valitsee sauvojen poikkileikkaukset. Ristikkojen topologiaa voidaan säätää topologiaparametreista.

Tässä esitettyjä ristikoita ei voitu toteuttaa luvussa 5.3.4 esitetyllä ristikkokomponentilla johtuen parametrien valinnasta. Kuten luvussa 4.3.2 on esitetty, algoritmisella mallilla voidaan tehdä ainoastaan sellaisia muutoksia, johon algoritmia luotaessa on varauduttu. Koska ristikkokomponentti on rakennettu siten, että sen avulla voidaan luoda ainoastaan

symmetrisiä harjaristikoita, ei sen muodostamalla paarteilla voida seurata muunlaista geometriaa. Näin ollen tarvitaan uusi algoritmi, joka muodostaa ristikoita geometriaparametreinaan nimenomaan annetut paarteiden tasot.

Algoritmisella suunnittelulla saavutetaan tässä lähtökohtaisesti nopeampi mallintaminen. Lisäksi algoritmiset ristikot seuraavat lähtötietona annettua geometriaa, jolloin mahdolliset suunnittelun edetessä tapahtuvat muutokset päivittyvät suunnitelmaan automaattisesti, kun parametreina olevat tasot päivitetään. Luotua algoritmia voidaan edelleen uudelleenhyödyntää tulevissa kohteissa. Uudelleenhyödyntämisen helpottamiseksi voidaan rakennetusta parametrisesta mallista luoda komponentti, jonka vapaita parametreja ovat riskon paarteiden ala- ja yläpinnat. Alkuperäinen algoritmi sekä vastaava komponentti ovat esitetty kuvassa 5.19.



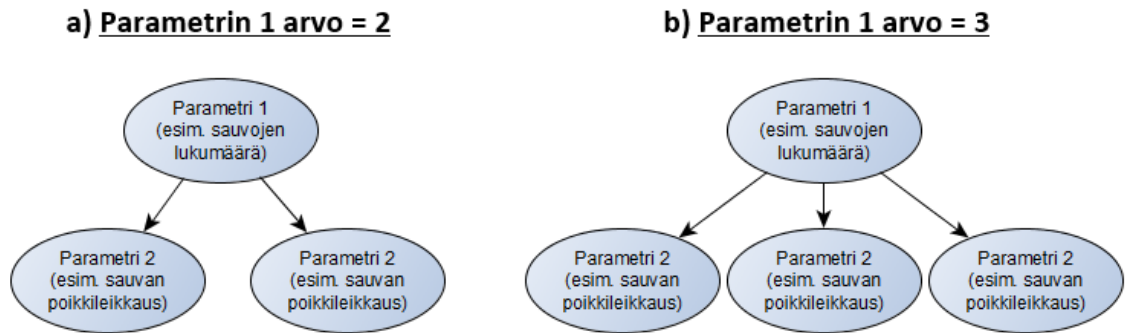
Kuva 5.19. Ristikkoalgoritmin kehitys komponentiksi.

Kuvassa 5.19 kohdassa a) on esitetty algoritmi, jolla myös kuvan 5.9 ristikot on muodostettu. Kohdassa b) algoritmista on muodostettu komponentti yleisiä pintoja varten. Kohdassa c) on komponentin avulla muodostettuja ristikoita muutetuilla tasoparametreilla.

Parametrisessa mallissa sekä komponentissa jokainen ristikko luodaan samoilla topologiaparametreilla. Tämä asettaa haasteita suunnittelualgoritmin kehittämiseksi, koska lähtökohtaisesti jokainen ristikko on erilainen. Näin ollen algoritmin tulee tunnistaa erot kunkin ristikon olennaisten rakenteellisten ominaisuuksien, kuten jännemitan ja rakennekor-

keuden välillä ja näiden perusteella käsitellä valittuja parametreja siten, että kustakin riskistä tulee järkevä. Vaihtoehtoisesti parametrit voitaisiin asettaa jokaiselle riskikollegalle erikseen. Tämä kuitenkin johtaa toisenlaisiin haasteisiin.

Tutkimuksessa havaittiin, ettei vapaiden parametrien lukumäärä saa riippua muiden parametrien arvosta. Kuvassa 5.20 on esitetty esimerkki tällaisesta tilanteesta.

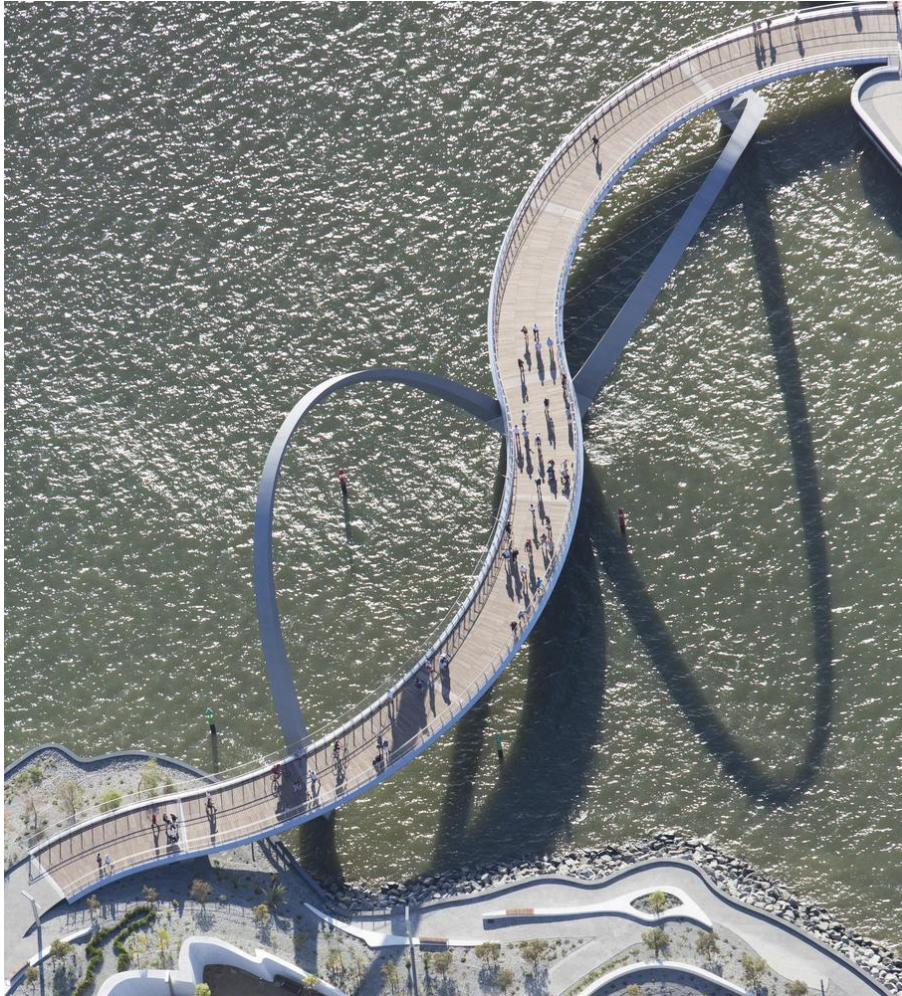


Kuva 5.20. Parametrimäärän riippuvuus toisesta parametrasta.

Kuvassa 5.20 on esitetty parametrin 2 riippuvuus parametrasta 1 kahdessa tapauksessa. Tapauksessa a) parametrin 1 arvo on 2, jolloin myös parametrien 2 lukumäärä on 2. Vastaavasti tapauksessa b) parametrin 1 arvo on 3, jolloin myös parametrien 2 lukumäärä on 3. Kun parametrin 1 arvoa, joka voi kuvata esimerkiksi rakenneosien lukumäärää, muutetaan, syntyy uusia parametreja tai vaihtoehtoisesti poistuu vanhoja. Mikäli parametri 2 on vapaa parametri, eli sitä ei ole lukittu algoritmilla, muuttuu mallin käsisyötteiden määrä. Koska uusia käsisyötteitä voidaan luoda malliin ainoastaan rajallinen määrä, aiheuttaa tämä ongelmia mallin toimivuudelle ja käytettävyydelle. Ongelma voidaan kiertää tapauksissa, joissa parametri 1 on rajattu pieneen, diskreettiin väliin, jonka jokainen mahdollinen arvo tunnetaan. Tällöin on mahdollista luoda parametrin jokaiselle arvolle omat syötteensä parametrille 2, jotka kytetään käyttöön parametrin 1 arvoista riippuvilla loogisilla porteilla. Tämä kuitenkin johtaa nopeasti mallia hallitsevien vapaiden parametrien määrän merkittävään kasvuun.

5.5.2 CASE: Elizabeth Quay Pedestrian and Cyclist Bridge

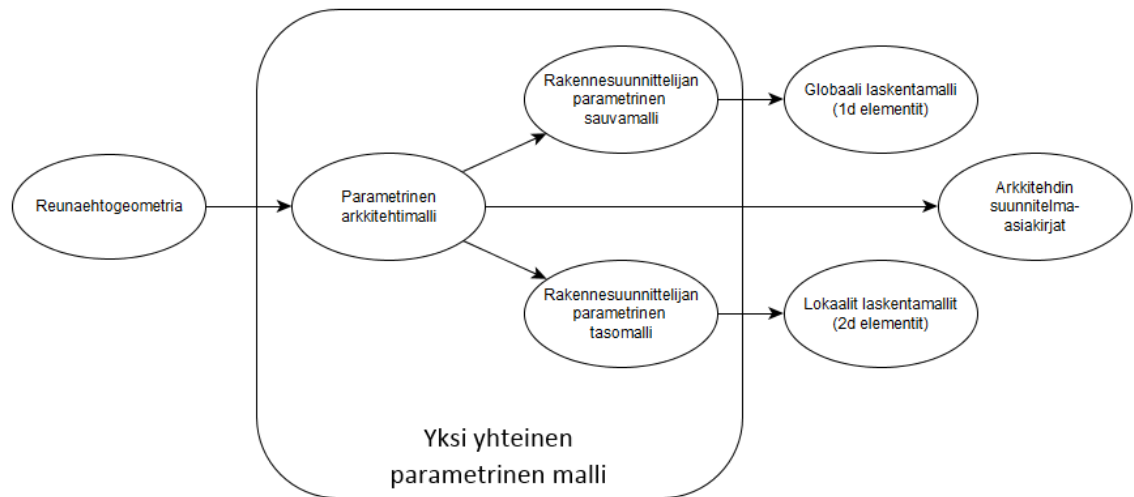
Elizabeth Quayn kevyenliikenteen silta on suunnittelutoimisto Arupin suunnittelema silta Australiassa, Perthissä. Sillan suunnittelussa on hyödynnetty parametrasta suunnittelua niin arkkitehtien kuin rakennesuunnittelijoidenkin osalta. Kuvassa 5.21 on esitetty ilma-kuva sillasta.



Kuva 5.21. *Elizabeth Quay Bridge (Archdaily)*

Kuvasta 5.21 havaitaan sillan monimutkainen geometria, jonka mallintaminen käsin rakennesuunnittelijan käyttöön tarkoitetuilla ohjelmistoilla olisi hankalaa. Lisäksi muodon etsiminen luonnosvaiheessa ja erilaiset muutokset työn edetessä aiheuttaisivat valtavan määrän käsityötä sillan geometrian mallintamisessa.

Sillan suunnittelussa käytettiin arkkitehdin ja rakennesuunnittelijan yhteistä geometriamallia. Malli muodostettiin yhdessä arkkitehdin ja rakennesuunnittelijan kanssa ottaen huomioon sekä tilaajan asettamat reunaehdot sillan geometrialle että rakennesuunnittelijan ja arkkitehdin asettamat reunaehdot sillan geometrian toteutuskelpoisuudelle. Yhteistyössä geometriaparametrit saatiin valittua siten, että sillan toteutus oli sekä arkkitehtonisesta että rakenteellisesta näkökulmasta mahdollista muuttamalla samoja parametreja. (Avern-Taplin et al. 2016, Summers & Buxton 2017)



Kuva 5.22. Parametrinen prosessi case-kohteessa (muokattu lähteestä (Avern-Taplin et al. 2016)).

Tässä projektissa rakenneanalyysit sekä dokumentaatio toteutettiin viemällä parametrisesti mallinnettu geometria vastaaviin suunnitteluohjelmiin kuvan 5.22 mukaisesti. Kuvasta nähdään suunnitteluprosessi geometrian luonnista lopullisiin suunnitelmiin. Ensimmäiset määräävät geometriset reunaehdot, kuten rantatörmät on mallinnettu kiinteästi. Tähän perustuen on luotu parametrinen malli geometriasta, joka sisältää kiinteitä reunaehtoja, muuttuvia reunaehtoja, sekä muutettavat parametrit. Parametrisesta geometriasta on edelleen luotu parametrisesti analyysimallit, jotka on siirretty FEM-ohjelmistoihin analysoitavaksi. Kun rakenne on suunniteltu arkkitehtonisesti ja rakenteellisesti tyydyttäväksi, on geometria siirretty dokumentointiohjelmistoihin toteutussuunnitelmien tuottamista varten. (Avern-Taplin et al. 2016)

Vertaamalla kohteen prosessikaaviota (kuva 5.22) aiemmin esitettyyn prosessikaavioon (kuva 5.16), havaitaan ettei tässä kohteessa suunnitteluprosessia ole täysin parametrisoitu. Rakennesuunnittelija saa lähtötietonsa parametrisesta arkkitehtimallista, mutta mikäli rakennesuunnittelu aiheuttaa muutostarpeita arkkitehdin suunnitelmiin, joudutaan nämä tekemään käsin. Näin ollen prosessi on perinteistä suunnittelumenetelmää muistuttava iteraatiivinen prosessi, jossa parametrista menetelmää hyödynnetään työmäärää pienentävänä työkaluna.

Sillan parametrista mallia ei tätä työtä tehtäessä ollut käytössä, joten seuraavat esimerkit eivät välttämättä vastaa todellista mallia, mutta auttavat ymmärtämään parametrisen mallin sisältöä. Kiinteä reunaehto voi olla esimerkiksi sillan vapaa alikulkukorkeus. Mikäli kulkuväylän sijaintia ei määrää mikään muu tekijä, kuten veden syvyys, voi se periaatteessa sijaita missä tahansa sillan jänteen kohdalla. Tilaaja on kuitenkin voinut määrätä vapaan korkeuden, jonka tulee täyttyä kulkuväylällä ja tämä on asetettava malliin reunaehdoksi. Muuttuva reunaehto voi olla esimerkiksi sillan kannen rakennekorkeus, joka

määritellään algoritmien perusteella siten, että se riippuu siihen rakenteellisesti vaikuttavista tekijöistä, kuten hyötyleveydestä ja köysivälistä. Parametreina voisi tässä mallissa olla esimerkiksi sillan kaarien korkeus ja kulma pystysuoraan nähden.

Huomattavaa on, että tällaisessa kohteessa reunaehtojen ja parametrien määrä on huomattavasti ylempänä esitettyä suurempi. Parametrien oikealla valinnalla ja järkevillä riippuvuussuhteiden ja reunaehtojen määrittämisellä voidaan merkittävästi vaikuttaa niin sillan ulkonäköön ja rakenteelliseen toimivuuteen kuin myös projektiin kuluvaan työmäärään. Näin ollen on syytä korostaa suunnittelun merkitystä ennen parametrin mallintamisen aloittamista.

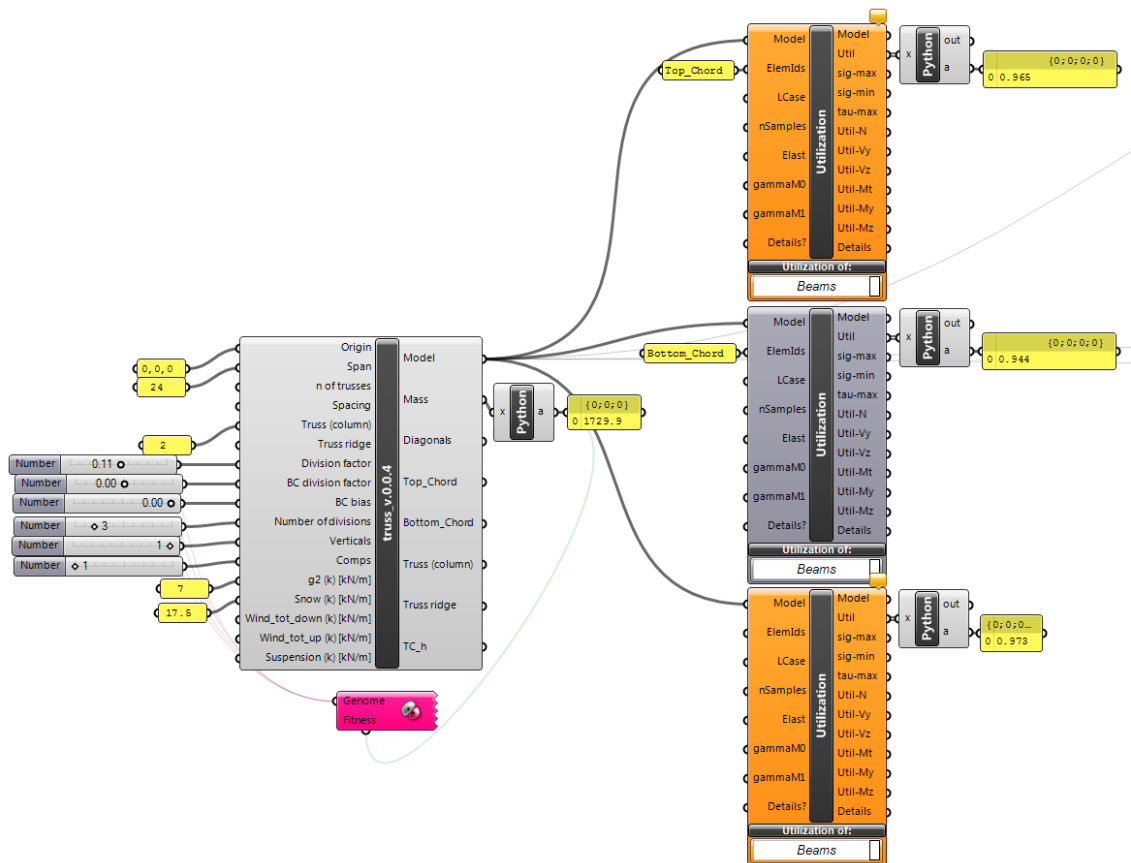
5.6 Optimointi

5.6.1 Tutkimusasetelma

Optimointi on mahdollista toteuttaa suuremmassa laajuudessa, esimerkiksi luvussa 5.4 esitellyn hallialgoritmin avulla koko rakennuksen laajuudelta. Tämä ei kuitenkaan ole tarkoituksenmukaista, koska monet rakennuksen geometrian ominaisuudet määräytyvät johtuen muista asioista, kuten arkkitehdista ja rakentamiskelpoisuudesta. Yksittäisten rakennosien, kuten pilarin tai palkin optimointiin taas ei tyypillisesti tarvita numeerisia optimointimenetelmiä, vaan paras ratkaisu saadaan läpilaskemalla parhaat vaihtoehdot. Näin ollen rakenteiden optimointia sovellettu parametriseen suunnittelumenetelmään tarkasteltiin ristikkokomponentin avulla, jonka topologia on tyypillisesti rakennesuunnittelijan määrättävissä.

Optimointi suoritettiin Grasshopperiin sisältyvällä Galapagos-optimointityökalulla. Rakennelaskelmat toteutettiin FEM-pohjaisesti Grasshopperin Karamba-lisäosalla. Vertailukohdaksi valittiin Melan (2013) väitöstyössään esittämä ristikon optimointi, joten koejärjestelyt toteutettiin soveltuvilta osin vastaavasti. Mela (2013) optimoi ristikkoo, jonka jänneväli on 24 m, alapaarten ja tukien välinen korkeus on 2 m, yläpaarten kulma on 1:20. Kuormana Mela (2013) on käyttänyt viivakuormaa 25,1 kN/m, joka sisältää lumi-kuorman ja katon omanpainon. Nämä kuormat yhdistellään eurokoodin mukaan. Koska kuormitustilannetta ei ole tarkemmin eritelty, saavutettiin tässä vastaava kuormitustilanne tarkastelemalla 7 m kehäjakoja omalla painolla 1 kN/m² ja lumikuormalla 2,5 kN/m². Näin saatiin viivakuormat 7 kN/m ja 17,5 kN/m vastaavasti, yhteensä 24,5 kN/m. Lisäksi laskentamalli ottaa huomioon ristikon omanpainon, jonka katsottiin täyttävän loput 0,6 kN/m. Kuormat yhdisteltiin eurokoodin (SFS-EN 1990 2002) mukaan. Terälaatuina käytettiin Melaa (2013) vastaavasti laatua S355. Ristikon sauvoina käytettiin SSAB neliöputkia, joista valittiin mukaan poikkileikkausluokat 1 ja 2.

Tutkimusasetelmassa tavoitefunktiona käytettiin ristikon massafunktiota, eikä erikseen määritettyä kustannusfunktiota. Lisäksi kapasiteettilaskelmissa jätettiin huomiotta teräsrakenteiden liitoseurokoodin (SFS-EN 1993-1-8 2005) mukaiset solmupistetarkastelut.



Kuva 5.23. Optimoitavien tutkimusasetelma.

Kuvassa 5.23 on esitetty optimoinnin tutkimusasetelma. Kiinteinä parametreina ovat edellä mainitut mitat ja kuormat. Optimoitavia parametreja on yhteensä kuusi. Parametri ”Division factor” määrää yläpaarteen solmupisteiden jakauman lineaarisesti siten, että arvolla 0 solmupisteiden etäisyys on jokaisessa välissä sama. Parametri ”BC division factor” valitsee alapaarteen solmupisteiden välin lineaarisesti siten, että arvolla 0 alapaarteen solmupiste on yläpaarteen projektion solmupisteiden puolella välissä. Positiivien arvo vetää solmupisteitä harjaa kohti ja negatiivinen puolestaan tukia kohti. ”BC bias”-parametrilla voidaan alapaarteen lineaarisesta solmuarvoa muuttaa toisen asteen funktioksi, jossa parametrin arvo vastaa toisen asteen termin kerrointa. Muutoksen suunta riippuu parametrin ”BC division factor” arvosta. ”Number of divisions” säätelee yläpaarteen solmuvälien lukumäärää. ”Verticals”-parametri voi saada arvot 0 tai 1, ja sillä valitaan lisätäänkö ristikkoon vertikaalit vai toimiiko se ilman vertikaaleja. ”Comps”-parametrilla valitaan kuinka monella tuelta päin laskettuna puristetulla sauvalla on eri poikkileikkaus kuin muilla uumasauvoilla. Ristikkokomponentissa on tässä tapauksessa rajoitettu mahdollisten uumasauvojen poikkileikkaukset kahteen. Ratkaisuna saadaan ristikon massa, paarteiden käyttöasteet sekä kokonaiskäyttöaste.

Optimointi suoritettiin molemmilla Galapagoksen tarjoamilla algoritmeilla: geneettisellä algoritmilla ja simuloidulla jäähtytyksellä. Molemmilla suoritettiin kolme erillistä sarjaa, jossa optimoitavien parametrien määrää muutettiin, yksi sarja kaikilla parametreilla, yksi

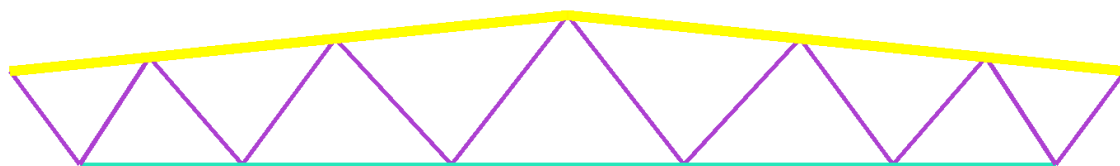
sarja ”BC bias”-parametrin arvolla nolla ja yksi sarja sekä ”BC bias” että ”BC division factor”-parametrien arvoilla nolla.

Kaikille optimointiajoille valittiin optimoitavien parametrien osalta sama lähtötilanne. Optimoitavat parametrit lähtötilanteessa on esitetty taulukossa 5.1.

Taulukko 5.1. *Optimoitavat parametrit.*

Parametrien lähtöarvot		
Parametri	Mahdolliset arvot	Lähtö-arvo
Division factor:	-2,50...2,50	-2,50
BC division factor:	-0,95...0,95	0,00
BC bias:	-1,00...0,00	0,00
Number of divisions:	2; 3; 4; 5; 6; 7	3
Verticals:	0; 1	0
Comps:	0; 1	1

Näillä parametreilla ristikon massaksi saatiin 2574,7 kg. Vastaava ristikko on esitetty kuvassa 5.24.



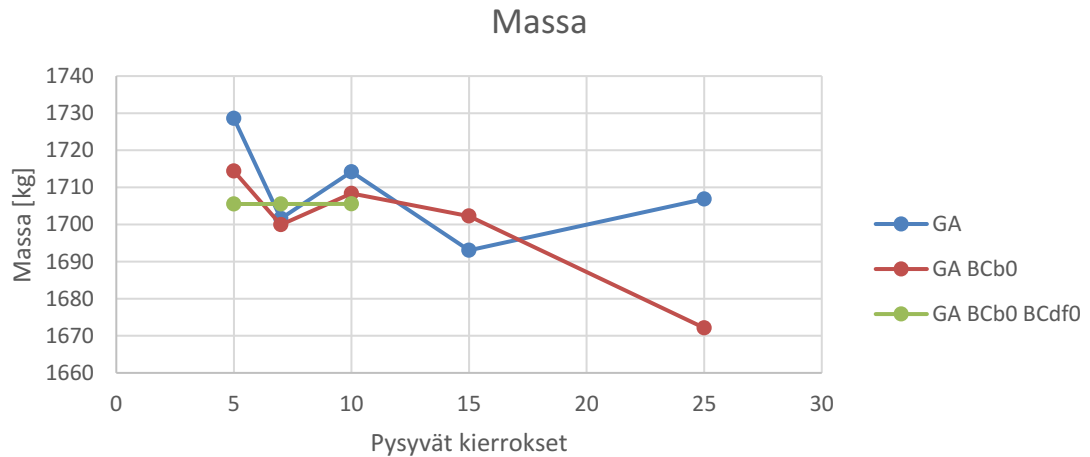
Kuva 5.24. *Optimoitavan ristikon lähtötilanne.*

5.6.2 Geneettinen algoritmi

Geneettisellä algoritmilla jokaisessa sarjassa suoritettiin viisi viiden optimointikierroksen ryhmää, joiden erona olivat vaadittu pysyvien kierrosten lukumäärä, joka tarkoittaa niiden kierrosten lukumäärää, joiden aikana tavoitefunktion arvo ei enää parane. Vakioidut Galapagoksen optimointiparametrit geneettisen algoritmin tapauksessa on esitetty taulukossa 5.2.

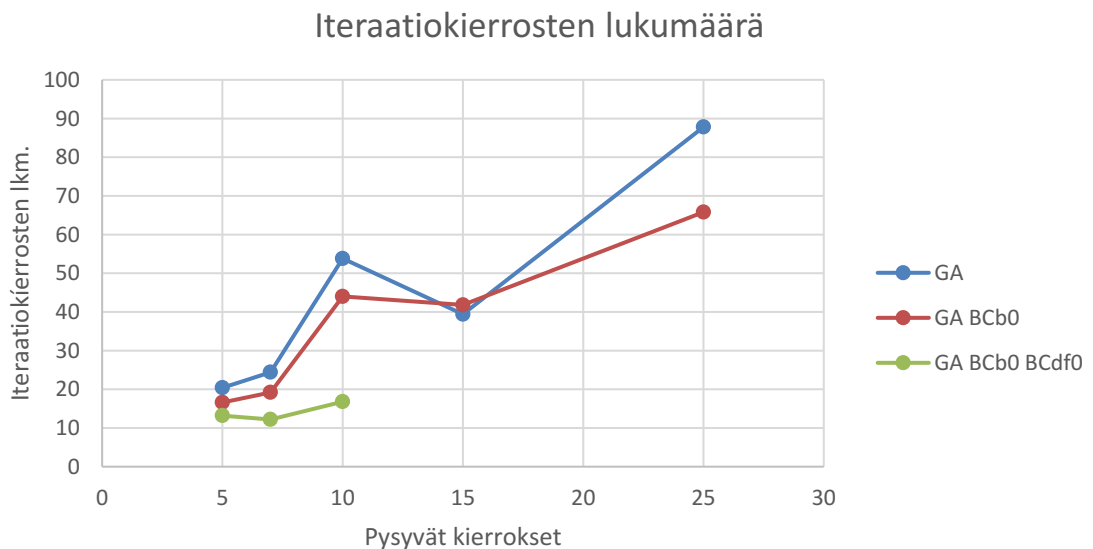
Taulukko 5.2. *Geneettisen algoritmin optimointiparametrit.*

Geneettinen algoritmi	
Population:	50
Initial boost:	2 x
Maintain:	10 %
Inbreeding:	+0.75 %



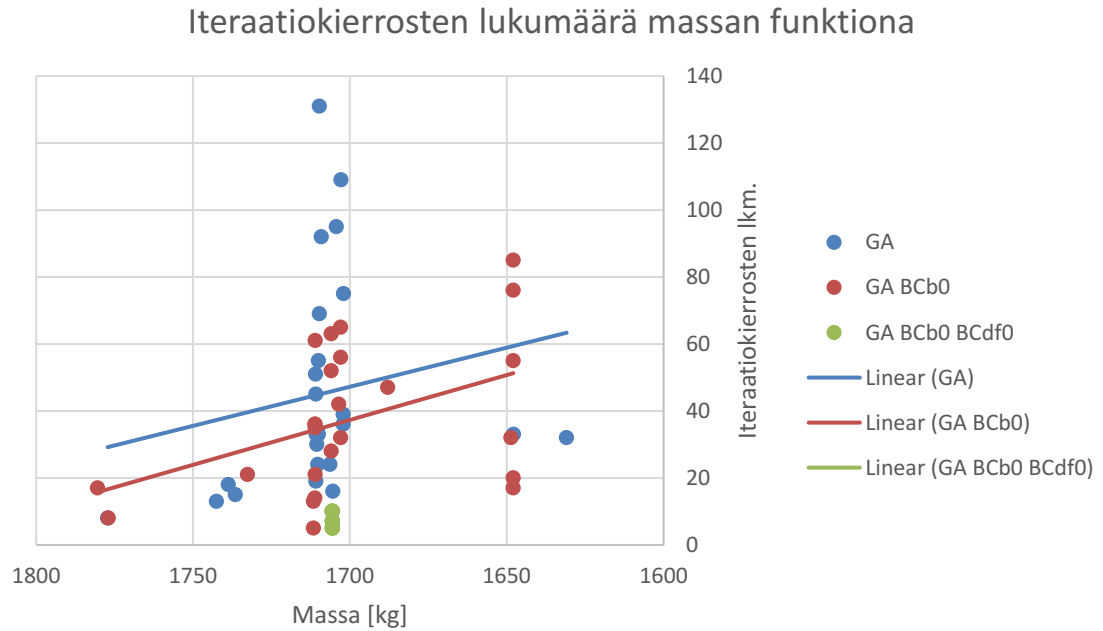
Kuva 5.25. Geneettisellä algoritmilla optimoitujen ristikoiden massojen keskiarvo pysyvien iteraatiokierrosten funktiona.

Optimoinnin tuloksena saadut kunkin optimointikierroksen ryhmän massojen keskiarvot pysyvien kierrosten funktiona on esitetty kuvassa 5.25. GA tarkoittaa ajoja, joissa optimoidaan kaikkia muuttujia, GA BCB0 ajoja, joissa parametrin ”BC bias” arvoksi on asetettu nolla ja GA Bcb0 BCDf0 ajoja, joissa parametrien ”BC bias” ja ”BC division factor” arvoiksi on asetettu nolla.



Kuva 5.26. Iteraatiokierrosten lukumäärä pysyvien iteraation funktiona (geneettinen algoritmi).

Optimoinnin tuloksena saadut kunkin optimointikierroksen ryhmän iteraatiokierrosten määrien keskiarvot pysyvien kierrosten funktiona on esitetty kuvassa 5.26. GA tarkoittaa ajoja, joissa optimoidaan kaikkia muuttujia, GA BCB0 ajoja, joissa parametrin ”BC bias” arvoksi on asetettu nolla ja GA Bcb0 BCDf0 ajoja, joissa parametrien ”BC bias” ja ”BC division factor” arvoiksi on asetettu nolla.

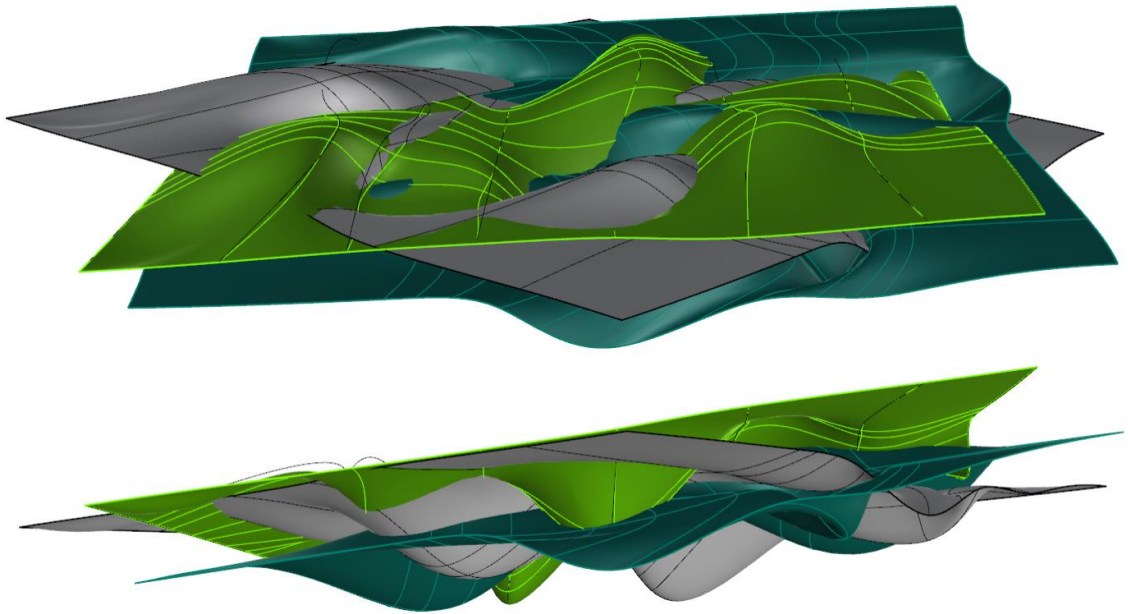


Kuva 5.27. Kaikkien geneettisen algoritmin optimointiajojen iteraatiokierrosten lukumäärä massan funktiona.

Kuvassa 5.27 on esitetty kaikkien optimointiajojen iteraatiokierrosten lukumäärä ristikon massan funktiona. GA tarkoittaa ajoja, joissa optimoidaan kaikkia muuttujia, GA BCb0 ajoja, joissa parametrin "BC bias" arvoksi on asetettu nolla ja GA Bcb0 BCdf0 ajoja, joissa parametrien "BC bias" ja "BC division factor" arvoiksi on asetettu nolla.

Jokaisessa kuvassa (kuvat 5.25...5.27) on esitetty viimeiseltä optimointiajokierrokselta tulokset ainoastaan pysyvien kierrosten määrillä 5, 7 ja 10. Tämä johtuu siitä, suuremmilla kierrosmäärillä algoritmi hukkasi löytämänsä optimin aina ennen vaaditun pysyvien kierrosten määrän täyttymistä, jolloin optimin etsiminen alkoi alusta, eikä algoritmi pysähtynyt. Näin ollen algoritmi pysäytettiin manuaalisesti eikä saatuja tuloksia kirjattu. Ongelma olisi korjattavissa korottamalla parhaiden tulosten säilytystodennäköisyyttä, mutta tutkimusasetelmaa ei tässä muutettu.

Geneettisellä algoritmilla suoritettujen optimointiajojen perusteella voidaan päätellä, että optimoitaviksi valituilla parametreilla on kriittinen merkitys niin optimoinnin tuloksen kuin kestonkin kannalta. Lisäksi lukituilla parametreilla on merkittävä vaikutus optimoinnin tuloksen kannalta. Tutkimusasetelman ja tulosten perusteella optimointitehtävästä voidaan piirtää skemaattinen kuvio kolmiulotteiseen avaruuteen, joka on esitetty kuvassa 5.28.



Kuva 5.28. Skemaattinen kuvio optimointitehtävän luonteesta.

Kuvassa 5.28 on esitetty erilaisia pintoja, joiden voidaan ajatella vastaavaan diskreettien muuttujien ("Number of divisions", "Verticals" ja "Comps") arvojen yhdistelmiä. Jatkuvat muuttujat puolestaan muodostavat näille pinnoille muotoja, mukaan lukien minimi ja maksimit. Eri tasoilla olevat ääriarvot voivat sijaita eri paikoissa, toisin sanoen jatkuvien muuttujien jollakin arvoilla voidaan toisella pinnalla saavuttaa minimi ja toisella väliarvoja tai jopa maksimi. Käytetty ratkaisija ei tallenna välituloksia optimoinnin vaiheista, mutta näyttää ne reaaliajassa visuaalisesti. Visuaalisen seurannan perusteella sekä tulosten (liite A ja kuva 5.27) voitiin päätellä, että löydettyt lokaalit minimi sijaitsevat suhteellisen kaukana toisistaan, koska ratkaisija löytää yleensä melko nopeasti (alle kymmenessä iteraatiokierroksessa), jonkin lokaalin minimialueen ja tämän jälkeen etsii sen minimiä. Tämä ominaisuus tosin parantui, kun optimoitavien parametrien määrää vähennettiin. Tämän selittämiseen on kaksi mahdollisuutta: parametrien määrän vähentäminen tuo lokaaleja minimejä lähemmäs toisiaan, jolloin algoritmi siirtyy helpommin minimialueesta toiseen tai toisaalta parametrien määrän vähennys saattaa algoritmista johtuen johtaa jäljelle jääneiden muuttujien suurempaan satunnaisuuteen.

Parhaat geneettisellä algoritmilla saavutetut tulokset ristikon massalle olivat luokkaa 1650 kg. Tyypillinen ratkaisu oli luokkaa 1700 kg. Tulosten perusteella geneettisellä algoritmilla on mahdollista löytää hyvä ratkaisu erityisesti, jos optimoitavien ja lukittujen parametrien valinta on suotuista. Erityisesti voidaan kiinnittää huomiota siihen, että optimointikierroksilla, joissa parametrin "BC bias" arvo lukittiin nolnaan, kävi sattumalta siten, että tämä arvo vastaa hyvin kaikissa optimointiajoissa löydettyä parasta ratkaisua. Tämä, sekä edellä mainittu minimien haun parametrien vähentäminen paransivat optimi-ratkaisun löytymisen todennäköisyyttä, kun tarkastelukierrosten lukumäärää kasvatettiin (kuva 5.27). On kuitenkin otettava huomioon, että paras ratkaisu (1630,9 kg), vaikkakin tässä tapauksessa suhteellisen pienellä massaerolla, löytyi kun optimoitavien parametrien

määrä oli suurin mahdollinen. Kun kaikki optimoitavat parametrit olivat mukana tarkastelussa, tulosten perusteella näytti siltä, että ratkaisun laatu perustuu ennemminkin jonkin lokaalin optimialueen löytymiseen, eikä iteraatiokierrosten määrän kasvattaminen parantanut tuloksia (kuva 5.25). Havaittiin myös, ettei geneettinen algoritmi näillä reunaehdoilla toimi kunnolla optimoitavien parametrien määrän ollessa pieni.

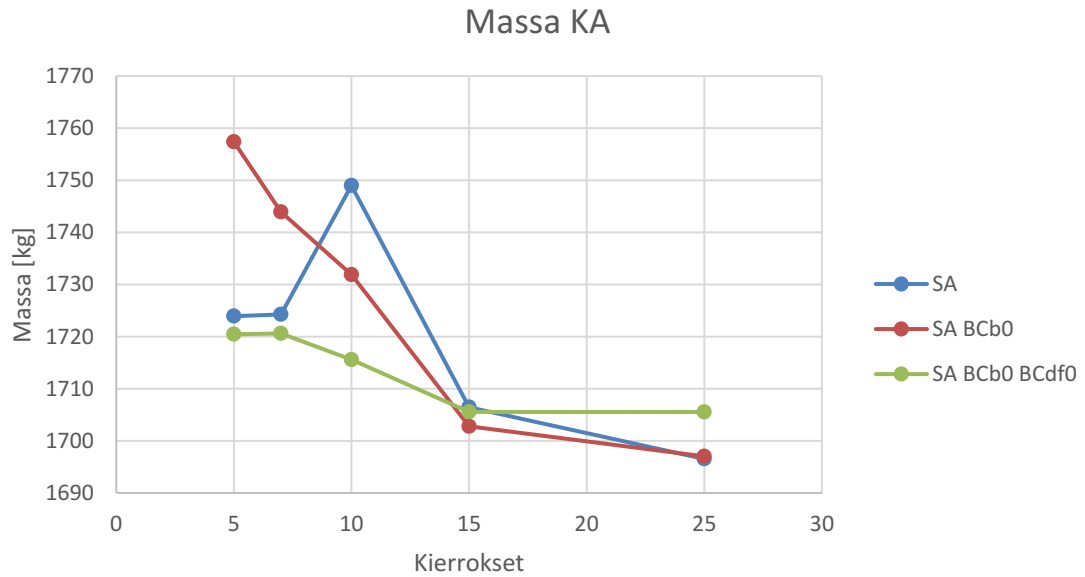
Optimointiajon pituutta säädettiin muuttamalla tuloksen osalta pysyväksi vaadittujen iteraatiokierrosten lukumäärää. Kuvasta 5.26 havaitaan, että tarvittujen iteraatiokierrosten lukumäärä kasvaa, kun pysyviä laskentakierroksia vaaditaan enemmän. Kuvasta 5.25 taas havaitaan, että paremman ratkaisun löytymisen todennäköisyys kasvaa iteraatiokierrosten määrä kasvaessa. Kuvasta havaitaan kuitenkin, että otoskoko on liian pieni ja päätelmä voidaan tehdä vain suuntaa-antavasti. Kuvan 5.25 perusteella voidaan väittää, että tutkimusasetelman tapauksessa hyvä ratkaisu (muttei välttämättä parasta), löydetään todennäköisesti, kun iteraatiokierrosten lukumäärä on suurempi tai yhtä suuri kuin 40. Kuvan 5.26 mukaan tämä saavutetaan, kun vaadittu pysyvien kierrosten lukumäärä on suurempi tai yhtä suuri kuin 10. Mikäli halutaan löytää paras mahdollinen ratkaisu, tarvitaan useita optimointiajoja näillä vaatimuksilla. Lisäksi havaittiin, että geneettisen algoritmin toiminta ja tehokkuus parantuu, kun optimoitavien parametrien lukumäärä kasvaa.

5.6.3 Simuloitu jäähdytys

Simuloidulla jäähdytyksellä jokaisessa sarjassa suoritettiin viisi viiden optimointikierroksen ryhmää. Ryhmien erona olivat yksittäisten ajojen lukumäärä, joka tarkoittaa kuinka monta erillistä optimointiajoa ratkaisija tekee. Vakioidut Galapagoksen optimointiparametrit on esitetty taulukossa 5.3.

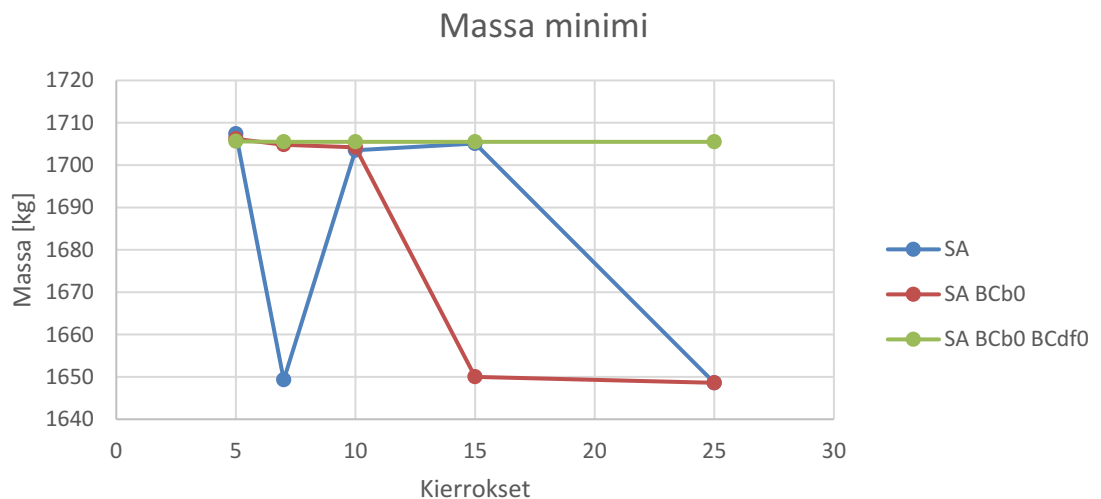
Taulukko 5.3. *Simuloidun jäähdytyksen optimointiparametrit.*

Simuloitu jäähdytys	
Temperature:	100 %
Cooling:	0.4 x
Drift Rate:	25 %



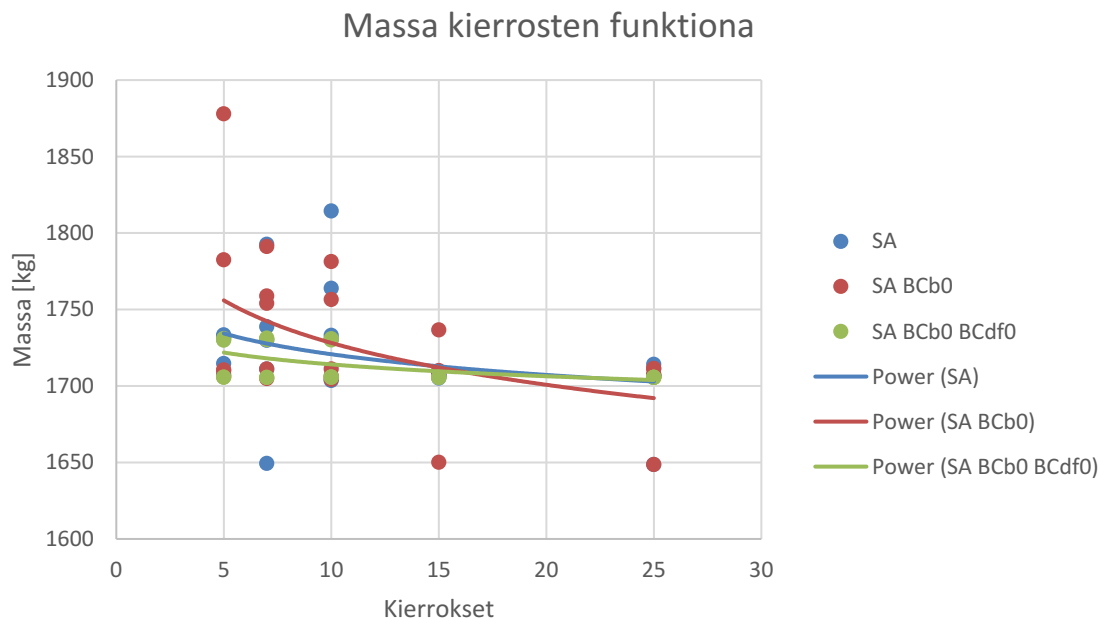
Kuva 5.29. Optimoitu massan keskiarvo simuloidulla jäähdytyksellä.

Optimoinnin tuloksena saadut kunkin optimointikierroksen ryhmän massojen keskiarvot on esitetty kuvassa 5.29. Koska simuloidun jäähdytyksen optimointikierroksen ryhmien yksittäiset optimointiajat ovat toisistaan riippumattomia, voidaan viiden optimointiajon ryhmän alkioden asemesta tarkastella koko ryhmän minimiä. Tässä tarkastelutavassa siis ryhmän jokaisen optimointiajon yhtä kierrosta tarkastellaan erillisenä, ja koko ryhmän tuloksista valitaan minimi. Vastaava tulos saataisiin ajamalla yhdellä optimointiajolla vastaava määrä kierroksia.



Kuva 5.30. Massojen minimi simuloidulla jäähdytyksellä.

Kuvassa 5.30 on esitetty massojen minimi. Lisäksi voidaan tarkastella optimoidun massan trendiä optimointiajokierrosten funktiona. Tällainen tarkastelu on esitetty kuvassa 5.31.

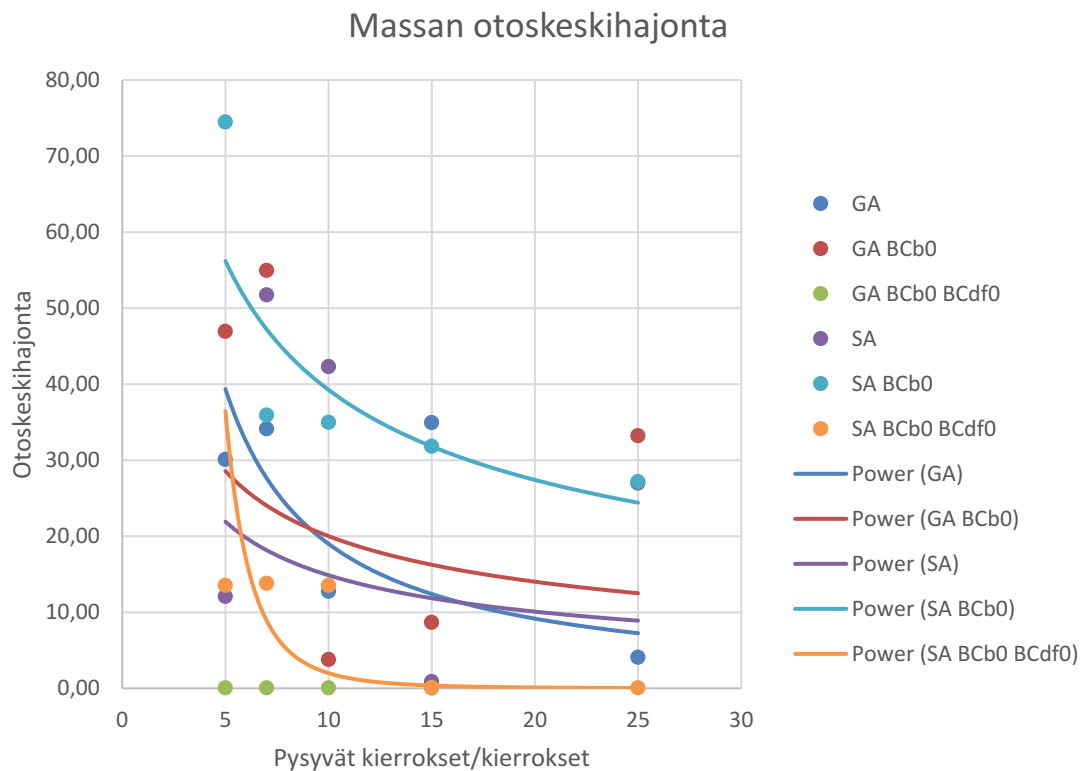


Kuva 5.31. Optimointiajokierrosten tuloksena saadut massat kierrosten funktiona.

Simuloidun jäähdtyksen osalta tulokset perustuivat satunnaisuuteen, jota voitiin hallita optimointiajojen määrällä. Kuvista 5.29 ja 5.30 nähdään, että optimoinnin luotettavuus sekä tulokset paranivat, kun optimointiajojen lukumäärää laskentakierroksessa kasvatettiin. Tämä johtuu puhtaasti todennäköisyydestä: kun optimointiajoja suoritetaan enemmän, on todennäköisempää löytää paras mahdollinen ratkaisu. Kuvan 5.31 perusteella voidaan todeta, ettei simuloitu jäähdtyks ole menetelmänä kovin herkkä parametrien määrälle. Hyvä ratkaisu löydetään, kun optimointiajossa suoritetaan riittävästi kierroksia. Tässä tapauksessa kierroksia tarvittiin 15 tai enemmän. Tutkimustapauksessa kuvan 5.30 perusteella optimointikierroksia tarvittiin yhteensä $5 \cdot 25 = 125$ kappaletta, jotta saavutettiin paras löydetty ratkaisu. Ei ole merkitystä suoritetaanko nämä erillisinä ajoina, vai yksittäisinä kierroksin yhdessä ajossa. Paras simuloidulla jäähdtyksellä saavutettu ristikön massa oli 1648,6 kg.

5.6.4 Yhteenveto

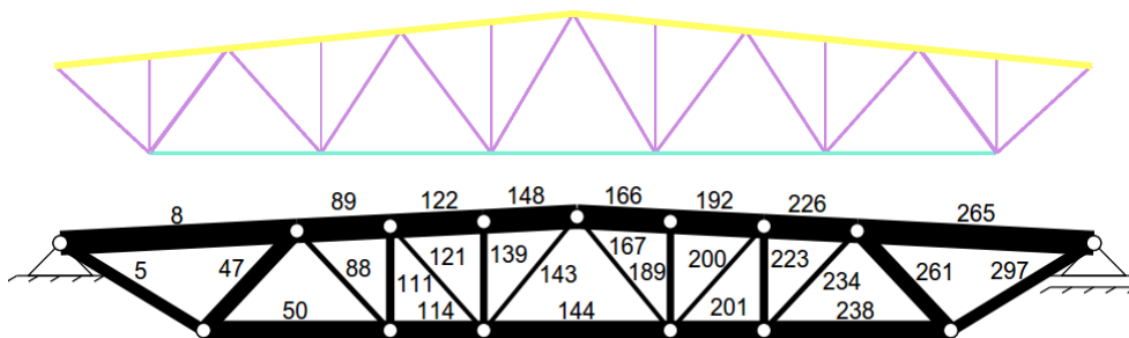
Kun optimointimenetelmiä vertaillaan keskenään, havaitaan, että simuloitu jäähdtyks soveltuu paremmin tarkasteltuun optimointitehtävään. Koska tehtävässä on huomattavasti epäjatkuvuutta ja joidenkin parametrien muuttaminen aiheuttaa merkittävää epäjatkuvuutta, kuten kuvasta 5.28 havaitaan, ei geneettisen algoritmin parhaista ominaisuuksista ole juuri hyötyä. Molemmat ratkaisumenetelmät löytävät aina jonkin lokaalin optimin, jolloin globaalien optimien löytäminen on kiinni todennäköisyydestä (eli lokaalien optimien kokonaismäärästä). Koska simuloitu jäähdtyks paikallisen haun menetelmänä suorittaa optimointikierroksen nopeammin kuin globaalien haun geneettinen algoritmi, voidaan samassa ajassa tarkastella enemmän ratkaisuja simuloidulla jäähdtyksellä.



Kuva 5.32. Optimoitujen ristikkojen massojen otoskeskihajonta.

Kuvassa 5.32 on esitetty optimoimalla saavutettujen massojen otoskeskihajonnat kierrosten funktiona molemmilla tarkastelluilla optimointimenetelmillä. Kuvasta havaitaan, että optimoinnin tulosten otoskeskihajonta pienentyi menetelmästä riippumatta, kun optimointiajojen lukumäärää kasvatettiin. Kuvasta on kuitenkin syytä ottaa huomioon, että kierrosten lukumäärä tarkoittaa eri menetelmillä eri asioita, kuten vastaavissa luvuissa on esitetty. Voidaan kuitenkin todeta, että optimoinnin luotettavuutta voidaan kasvattaa optimointikierrosten lukumäärää kasvattamalla.

Optimoinnin paras tulos ei menetelmästä riippuen juurikaan vaihdellut (geneettinen algoritmi: 1630,9 kg ja simuloitu jäähdytys: 1648,6 kg). Paras tulos saavutettiin siis geneettisellä algoritmilla ja sitä vastaava ristikko on esitetty kuvassa 5.33.



Kuva 5.33. Optimoitu ristikko sekä vertailuristikko (Mela 2013).

Vertailun vuoksi kuvassa 5.33 on esitetty myös Melan (2013) optimoinnin tuloksena saavuttama ristikko, jonka massaksi oli saatu 1037,9 kg. Vaikka Mela (2013), oli määrittänyt tavoitefunktion tässä esitettyä tarkastelua tarkemmin, ei sitä voida pitää selityksenä erolle. Erona ei myöskään voida pitää käytettyjä optimointimenetelmiä, koska molemmilla menetelmillä saavutettiin samankaltaisia eroja. Suurimmat erot massojen välillä syntyvät nimenomaan optimointitehtävän asettelusta. Melan (2013) optimointitehtävä oli aseteltu siten, että ristikolla on määrätty määrä solmupisteitä, joiden väliin luodaan sauvat tietyin reunaehdoin kuitenkin siten että sauvat saavat lävistää toisensa. Tästä perusrakenteesta alettiin optimointimenetelmällä poistaa sauvoja siten, että jäljelle jäävä rakenne on mahdollisimman edullinen. Tämän tutkimuksen optimointitehtävän asettelu taas vastaa selkeästi enemmän rakennesuunnittelijan toimintatapaa ristikkoa suunniteltaessa. Ristikon topologialle annetaan jonkinlainen alkuarvaus perustuen suunnittelijan kokemukseen. Alkuarvausta lähdetään kehittämään siirtämällä solmupisteitä suunnittelijan ohjeistustutkimuksen perusteella siten, että sauvojen poikkileikkauksia voidaan pienentää. Rakennesuunnittelijan iteraatiokierroksia rajoittaa käytettävissä oleva aika ja lujuuslaskennan työläys. Tässä esitetty optimointimenetelmä taas siirtelee solmuja, kunnes löytää niille optimaalisen sijainnin tavoitefunktion kannalta. Näin ollen solmujen asemia säätelevillä parametreilla on olennainen asema, kun tarkastellaan parhaan mahdollisen ratkaisun saavuttamista. Tässä parametrit oli määriteltä siten, ettei vertailuristikon kaltaista ristikkoa, jossa on N-liitoksia, ollut mahdollista saavuttaa, mikä johtaa myös suurempaan ristikon massaan.

5.7 Taloudellisuus

Rakenneosien tarkastelun yhteydessä luvuissa 5.3.2 ... 5.3.6 esiteltiin komponenttipohjainen tapa parametriseen rakennesuunnitteluun. Komponenttien tarkoituksena on säästää aikaa parametrinen mallien luomisessa ja näin parantaa parametrinen mallintamisen kannattavuutta. Tutkimuksessa havaittiin, että komponenttien hyödyntämisestä mallintamisessa saavutetaan eniten hyötyä, kun niitä hyödynnetään monimutkaisempien rakenneosien, kuten ristikoiden ja liitosten mallintamisessa. Yksinkertaisempien rakenneosien, kuten teräksisen palkin ja pilarin, parametrinen mallintaminen on lähtökohtaisesti niin

yksinkertaista, että komponentin hyödyntämisestä saatava etu jää pieneksi. Monimutkaisempien rakenneosien tapauksessa taas komponenttia vastaavan algoritmin luomisessa saattaa kestää jopa kymmeniä tunteja, jolloin komponentin hyödyntäminen muodostuu taloudelliseksi, koska se onnistuu hyvin luotujen ja soveltuvien komponenttien osalta muutamassa minuutissa. Toisaalta voidaan todeta, että tutkimuksessa rajattiin pois alustavien tarkastelujen jälkeen teräsbetonirakenteet erityisesti johtuen tiedonsiirron haasteista eri ohjelmistojen välillä. Mikäli tämä haaste saadaan tulevaisuudessa ratkaistua, voidaan tämän työn perusteella esittää, että teräsbetonirakenteet soveltuvat erityisen hyvin komponenttipohjaiselle suunnittelutavalle. Tämä johtuu siitä, että teräsbetonirakenteissa parametrimäärä on huomattavasti suurempi kuin homogeenisissa materiaaleissa johtuen raudoituksesta ja sen sijoittelusta.

Tutkimuksessa tarkasteltiin taloudellisuutta myös aikaperusteisesti. Luvussa 5.4 esitellyn hallialgoritmin luomiseen kului aikaa yhteensä 80 tuntia. Vaikka algoritmista on puutteita, eikä sillä voida luoda täysin mielivaltaisia rakennuksia, voidaan algoritmilla kuitenkin tuottaa tehokkaasti luonnossuunnitteluvaiheeseen riittäviä suunnitelmia. Lisäksi säästetään nopeasti tieto rakennuksen rungon kustannuksiin olennaisesti vaikuttavista tekijöistä.

Molemmissa edellä mainituissa tapauksissa rakennesuunnittelijan kannalta keskeinen taloudellisuuden näkökulma on uudelleenhyödyntäminen. Komponenttien uudelleenhyödyntämisellä lyhennetään parametriseen suunnitteluun kuluva aikaa. Tämä parantaa parametriseen suunnittelutavan kilpailukykyä verrattuna perinteiseen suunnittelutapaan, koska lähtökohtaisesti parametriseen suunnitteluun kuluu perinteistä suunnittelumenetelmää enemmän aikaa ja parametrissa suunnittelutapaa puoltavat tekijät on löydettävä luvussa 4.3.1 esitetyistä tekijöistä suunnittelutapaa valittaessa. Kokonaisten algoritmien, kuten esitetyn hallialgoritmin uudelleenhyödyntämisen edut perustuvat samankaltaisten kohteiden suunnitteluun, jossa algoritmia voidaan joko suoraan tai pienin muutoksin hyödyntää kohteen suunnittelussa. Suoraan sopivalla, hyvin rakennetulla algoritmilla voidaan kohde teoriassa suunnitella muutamassa minuutissa mukaan lukien lujuuslaskenta ja mallinnus. Käytännössä tilanne ei yleensä ole näin optimaalinen, mutta esimerkiksi erilaisissa konseptirakennuksissa tai -rakennelmissa, joita tuotetaan useita vastaavia ennalta määrätyillä vapailla parametreilla, tätä päästään hyödyntämään. Tällaisissa rakenteissa mallin parantamisen ja kehittämisen hyödyt kumuloituvat aina seuraavien suunnittelutehtävien lyhenemisen muodossa.

Toinen rakennesuunnittelijan kannalta olennainen hyötynäkökulma on mallintamisen nopeus erityisesti geometrisesti monimutkaisten rakenteiden osalta. Luvussa 5.5.1 esitetyn kaarevia ristikoita tuottavan algoritmin luomiseen kului ristikoiden geometrian osalta noin kolme tuntia, kun parametreina toimivien tasojen geometria oli tiedossa. Vastaavan geometrian luomiseen kuluva aika mallintamalla suunnitteluohjelmistolla kuten Tekla Structuresilla, voi vaihdella muutamasta tunnista useisiin kymmeniin tunteihin riippuen

lähtötietotasojen geometrian aiheuttamasta eroista ristikoille sekä mallinnettavien ristikoiden määrästä. Edelleen parametrisella mallilla voidaan tehdä nopeasti muutoksia ja siihen voidaan tarpeen mukaan liittää lujuuslaskentaominaisuuksia tehokkuuden kasvattamiseksi.

Muiden rakennusprojektin osapuolten kannalta parametrinen suunnittelun taloudellisuus perustuu nopeaan vaihtoehtojen tuottamiseen ja näiden hyötyjen ja haittojen selkeään ilmaisuun niin kustannuksiin vaikuttavien tekijöiden osalta kuin ratkaisujen visualisointiin kolmiulotteisessa, nopeasti ymmärrettävässä muodossa. Nämä tekijät auttavat tilaajaa tekemään tarpeisiinsa ja resursseihinsa nähden parhaan mahdollisen valinnan huomattavan pienillä esisuunnittelukustannuksilla. Käytännössä tämä tarkoittaa, että tilaaja voi vastaavalla rahalla vertailla useampia vaihtoehtoja käytettäessä parametrista menetelmää kuin käytettäessä perinteistä suunnittelumenetelmää. Toinen merkittävä hyöty tilaajan näkökulmasta on materiaalinkäytön optimointi. Kuten luvussa 5.6 on esitetty, parametrinen suunnittelutapa helpottaa optimointialgoritmien hyödyntämistä suunnittelutehtävien yhteydessä. Kun optimointi integroidaan osaksi parametrista suunnitteluprosessia, rakenteista suunnitellaan perinteistä suunnittelutapaa tehokkaampia ja kevyempiä, mikä johtaa taloudellisuuteen tilaajan näkökulmasta.

6. YHTEENVETO

6.1 Parametrisen suunnittelun hyödyt ja haasteet

Parametrisessa suunnittelussa rakenteen suunnitellaan algoritmien avulla. Algoritmit käyttävät lähtötietoinaan sekä avoimia että lukittuja parametreja, muodostaen parametrisen mallin. Mallin avulla muodostetaan suunnitteluratkaisu perustuen parametrien arvoihin. Parametrisella suunnittelumenetelmällä voidaan saavuttaa merkittäviä hyötyjä verrattuna perinteisiin suunnittelumenetelmiin. Tärkeimmät parametrisella suunnittelulla tai mallintamisella saavutettavat hyödyt ovat:

- perinteistä mallintamistapaa nopeampi geometrisesti monimutkaisten rakenteiden mallintaminen
- periaatteeltaan samankaltaisten rakenteiden tai rakennusten suunnittelu sarjatyönä
- nopea muutosten toteutus
- erilaisten suunnitteluratkaisujen tehokas vertailu.

Parametrinen suunnittelu on järkevää, kun mallinnettavana on useita geometrisesti hankalia rakenneosia. Olennaista on, että parametrisen mallin luomisen on oltava nopeampaa kuin rakenneosien mallintaminen käsin tai että parametrisesta suunnitteluprosessista saavutetaan muita etuja mallinnustyön lisäksi. Suunnittelu on järkevää toteuttaa parametrisesti myös silloin, kun odotettavissa on paljon muutoksia. Tämä pätee kuitenkin vain silloin kuin muutosten laatu on odotettavissa, jolloin parametrinen malli voidaan suunnitella ja luoda siten, että muutokset voidaan toteuttaa auki jätettyjen parametrien arvoja muuttamalla. Vastaavalla mallilla on myös nopea suunnitella samankaltaisia rakenteita tai rakennuksia. Neljäs hyötynäkökulma, erilaisten suunnitteluratkaisujen vertailu, on esitetyistä hyödyistä helpoiten ennustettavissa ja toteutettavissa. Koska esisuunnitteluvaiheessa tiedetään, että halutaan vertailla erilaisia rakenteellisia vaihtoehtoja, on tarve parametriselle suunnittelulle lähtökohtaisesti olemassa. Koska kohteelle on yleensä määritellyt tavoitteet, voidaan luoda parametrinen malli, jonka lopputulos saavuttaa nämä tavoitteet riippumatta parametrien arvoista (kun parametrien arvojoukkoa on rajoitettu). Mikäli erilaiset tarkasteltavat vaihtoehdot ovat järkevästi eriteltävissä jonkin parametrin tai parametrien arvoilla (esimerkiksi jänneväli), on suunnittelu järkevää toteuttaa parametrisesti.

Parametrisen suunnittelun kannalta suurin ongelma on edellä mainittujen hyötyjen realisoitumisen ennustaminen. Rakennesuunnittelijan kannalta kaikki parametrisen suunnittelun edut realisoituvat valmiin kohteen perinteistä suunnittelumenetelmää lyhyempänä suunnitteluajana. Koska suunnittelutapa on suunniteltavan rakennuksen tai rakenteen osalta valittava ennen suunnittelua, tarvitaan luotettava arvio suunnitteluun kuluvalle

ajalla molemmilla suunnittelumenetelmillä. Suunnitteluun kuluva ajasta on suunnittelutoimistoilla olemassa kokemuseräistä tietoa perinteisen suunnittelumenetelmän osalta, mutta parametriseen suunnittelun osalta kokemuseräinen tieto puuttuu. Koska parametriseen suunnitteluun kuluva ajan arviointiin ei ole olemassa kokemuseräistä tietoa, joudutaan suunnitteluun kuluva aika arvaamaan ilman vertailukohtia. Tämä puoltaa perinteisen suunnittelumenetelmän valintaa, koska luotettavan työmääräarvion tekeminen on olennaista liiketoiminnan kannalta.

Parametrisella mallilla suunnitelmiin ei voida mielivaltaisia muutoksia, vaan ainoastaan sellaisia muutoksia, joita luotu parametrinen malli sallii. Näin ollen myös tulevat muutokset tai mahdolliset muutostarpeet tulee pystyä ennustamaan mallin suunnitteluvaiheessa. Mallia ei ole yleensä järkevä rakentaa siten, että se mahdollistaa kaksi tai useampia täysin eri rakennemalleja, ellei ole selvää näyttöä siitä, että kaikkia mahdollistettuja vaihtoehtoja tullaan tarvitsemaan. Näin ollen mahdollisten muutostarpeiden huomioon ottaminen tulee tehdä parametriseen mallin suunnitteluvaiheessa samassa yhteydessä, kun päätetään kohteen parametrisesta suunnittelusta.

Muita parametriseen suunnitteluun liittyviä haasteita ovat erityisesti osaamisen tarve. Parametriseen suunnittelutavan opetus kouluissa on vähäistä. Näin ollen kaikilta rakennesuunnittelijoilta, jotka eivät ole toimintatapaa itse opiskelleet, puuttuu lähtökohtaisesti parametriseen suunnitteluun soveltuva ajattelumalli sekä menetelmien osaaminen. Lisäksi parametrisissa malleissa vapaaksi jätettävien parametrien valinta saattaa joissakin tilanteissa olla vaikeaa. Vapaaksi jätettävien parametrien määrään vaikuttavat mallin tavoitteiden lisäksi rakennososan optimointitarpeet sekä algoritmin tavoiteltu uudelleenhyödynnettävyys. Algoritmin on rakennesuunnittelijan näkökulmasta helppokäyttöisin ja tehokkain kun vapaiden parametrien lukumäärä on mahdollisimman pieni mallille asetettujen ensisijaisten tavoitteiden saavuttamiseksi. Lisäparametreilla voidaan saavuttaa lisäominaisuuksia, mutta ne myös kasvattavat mallin luomiseen tarvittavaa työmäärää. Mallin tärkeimmät vapaat parametrit on aina suunniteltava ja sidottava tavoitteisiin.

Tutkimuksessa havaittiin, että kantavien rakenteiden parametriseen suunnittelun vaikeusaste on riippuvainen rakennusmateriaalista. Tämä johtuu suoraan mahdollisten parametrien lukumäärästä eli materiaalin parametripuusta. Teräksellä mahdollisten parametrien lukumäärä on pieni, kun taas teräsbetonilla huomattavan suuri. Parametrimäärän kasvu johtaa suoraan parametriseen mallin rakentamisen monimutkaistumiseen, sillä jokaiselle parametrille on suunnittelussa määrättävä arvo. Toisaalta monimutkaisten parametrusten mallien uudelleenhyödyntäminen esimerkiksi komponenttipohjaisesti mahdollistaa merkittävästi perinteistä suunnittelumenetelmää tehokkaamman suunnittelun.

Toteutuessaan parametrisesta suunnittelumenetelmästä saavutettavat edut realisoituvat rakennesuunnittelijalle lyhyempänä suunnitteluun kuluvana aikana. Parametrisella menetelmällä suunnittelija pystyy automatisoimaan osan tehtävistään ja esimerkiksi muutos-

tenhallinta nopeutuu. Vaikka työssä pyrittiin löytämään parametrinen suunnittelun hyötyjä erityisesti rakennesuunnittelijan näkökulmasta, havaittiin että tilaajan saavuttamat hyödyt ovat varmemmin saavutettavissa, koska niin ei liity vastaavaa epävarmuutta kuin rakennesuunnittelijan hyötyihin. Tilaajan kannalta keskeisiä parametriseista suunnittelumenetelmästä saavutettavia etuja ovat perinteistä laajempi erilaisten vaihtoehtojen vertailu, sekä parametrisein menetelmän mahdollistama, resurssien kannalta tehokas, optimointi. Vaihtoehtojen vertailun taloudelliset hyödyt riippuvat tilaajan tavoitteista. Mikäli vertailua käytetään erilaisten ratkaisujen rakentamiskustannusten vertailuun, ovat hyödyt suoraan taloudellisia. Muussa tapauksessa hyödyt realisoituvat mulla tavoin, esimerkiksi tilojen toiminnallisuuden parantumisena, eivätkä näin ollen ole suoraan mitattavissa. Myös optimoinnista saavutettavat hyödyt hyödyttävät tilaajaa suoraan taloudellisesti pienempinä rakentamiskustannuksina. Voidaan myös ajatella, että mikäli mainitut edut tilaajalle pystytään osoittamaan suunnittelun tarjousvaiheessa, voi suunnittelija saavuttaa tällä kilpailuetua verrattuna kilpailijoihin, jotka eivät vastaavaa menetelmää hyödynnä.

6.2 Parametrisein mallin suunnittelu, toteutus ja optimointi

Tutkimuksen perusteella voitiin osoittaa, että parametrisein malli tulee aina suunnitella. Suunnittelun tarve on suora seurausta siitä, että parametrisein malli tekee ainoastaan ne asiat, jotka siihen on sisällytetty. Suunnittelemalla varmistetaan mallin käyttökelpoisuus, toteuttamiseen tarvittava työmäärä sekä mahdollisuus hyödyntää mallia tulevaisuudessa. Näin ollen ennen parametrisein mallin aloittamista tulee suunnitella ainakin mallin käyttötarkoitus mukaan lukien olennaisimmat parametrit, joiden avulla tavoitteet ja käyttötarkoitus saavutetaan, sekä tieto, jota mallin tulee tuottaa. Lisäksi pitää tarkastella millaisia ohjelmistoja suunnittelutyössä käytetään ja mitkä ovat liittymäpinnat muiden suunnittelualojen kanssa. Myös uudelleenhyödyntämistä, niin vanhojen mallien hyödyntämistä uuden mallin luonnissa, kuin uuden mallin hyödyntämistä tulevissa malleissa tulee tarkastella, jotta parametrisein malli on mahdollista toteuttaa taloudellisesti.

Olennaisia mallin toteutuksessa huomioon otettavia tekijöitä ovat tarkoituksenmukaisuuden lisäksi taloudellisuus eli toteuttamiseen kuluva aika sekä ymmärrettävyys eli se että mallia voivat hyödyntää ja muokata useat henkilöt. Näiden tekijöiden saavuttamiseksi työssä on esitetty komponenttipohjainen tapa toteuttaa parametriseia malleja. Komponenttipohjaisessa mallintamistavassa rakenneosia, tai muita vastaavia selkeitä kokonaisuuksia varten toteutetaan parametrisein komponentti. Komponentin avulla (sen parametrisein arvoja muuttamalla) voidaan parametrisein malliin luoda haluttu, komponenttia vastaava rakenneosa. Koska komponentissa rakenneosan parametrisein malli ja sen ominaisuudet on luotu valmiiksi helposti hyödynnettäväksi kokonaisuudeksi, nopeuttaa menetelmä parametrisein mallin toteuttamista, kun käyttäjän ei tarvitse toistuvasti uudelleen miettiä rakenneosan parametrisein mallintamistapaa ja toteuttaa sitä. Komponenttipohjainen tapa myös selkeyttää parametrisein mallia, koska kukin komponentti tekee ainoastaan sille mää-

rätyn tehtävän, jolloin myös muiden käyttäjien on helppo ymmärtää sen lähtötietoparametrit ja jatkokäsittellä tulokset. Tutkimuksessa havaittiin, että komponenttipohjainen mallintamistapa soveltuu erityisen hyvin monimutkaisiin rakennesein, joiden geometrian määrittämiseen sekä lujuuslaskentaan vaaditaan merkittävä määrä algoritmisia yhteyksiä ja laskutoimituksia. Tällaisia ovat työssä tarkastelluista rakenneseistä erityisesti ristikot ja liitokset. Yksinkertaisten rakenneseiden osalta saavutettavat hyödyt ovat suhteelliseen pieniä suoraan parametriseen mallintamiseen verrattuna.

Parametrinen mallintamistapa soveltuu hyvin myös rakenteiden optimointiin, koska mallintamistavassa rakenteiden ominaisuuksia jätetään muuttujilla säädettäviksi parametreiksi. Nämä parametrit soveltuvat joko suoraan tai pienillä muutoksilla optimointitehtävän muuttujiksi. Rakenneseiden optimointi joudutaan parametrisia suunnittelumenetelmiä hyödynnettäessä tekemään tyypillisesti numeerisilla optimointimenetelmillä. Tutkimuksessa verrattiin parametrisen ristikon optimointia kahdella eri numeerisella menetelmällä: geneettisellä algoritmilla sekä simuloitulla jäähdytyksellä. Tulosten perusteella vaikutti siltä, että simuloitu jäähdytys on parempi optimointimenetelmä erityisesti optimoitavien parametrien määrän kasvaessa. Optimoinnilla saavutettiin massassa mitattuna säätöjä verrattuna tavanomaiseen suunnittelumenetelmään. Lähdekirjallisuudessa esitettyjä optimointituloksia ei kuitenkaan saavutettu. Kuitenkin optimointi havaittiin nopeutensa ja tulostensa vuoksi hyödylliseksi menetelmäksi ja sitä suositellaan sovellettavan parametrisen suunnitteluprosessin yhteydessä.

6.3 Tutkimustarpeet tulevaisuudessa

Tässä työssä on esitelty periaatteet ja mahdolliset toimintatavat kantavien rakenteiden parametriseen mallintamiseen niin rakennesuunnittelijan osalta kuin rakennesuunnittelijan ja arkkitehdin yhteistyön osalta. Lisäksi työssä on esitetty parametriseista suunnittelusta saavutettavat hyötynäkökohdat erityisesti rakennesuunnittelijan osalta. Seuraavassa on esitelty asioita, joita tämän tutkimuksen perusteella tarvitsevat lisäselvityksiä.

Parametrisein suunnittelun heikkoudeksi tunnistettiin kokemuseräisen kustannustiedon puute, mitä on olemassa perinteisellä suunnittelumenetelmällä toteutetuista kohteista. Kustannustiedon puutteen vuoksi suunnittelumenetelmiä on vaikea vertailla keskenään, ja näin ollen on vaikea osoittaa parametriseista menetelmästä saavutettavia hyötyä. Edelleen on vaikea luotettavasti osoittaa ja tehdä päätöksiä siitä, minkälaisissa kohteissa parametriseista suunnittelua olisi järkevä soveltaa. Mikäli parametrisein suunnittelun hyödyt ja soveltuvuus erilaisiin kohteisiin halutaan osoittaa, tarvitaan tutkimustietoa menetelmän kustannuksista.

Eräs tutkimuksessa esitetty parametriseista suunnittelusta saavutettava hyöty on mahdollisten muutosten tehokas toteuttaminen perinteistä suunnittelumenetelmää pienemmällä virheiden riskillä. Toisaalta esitetään, että parametrinen malli pystyy toteuttamaan aino-

astaan sellaisia muutoksia, jotka on mahdollistettu mallin luontivaiheessa. Lähtökohtaisesti tarvittavat suunnitelmamuutokset rakennesuunnitteluprosessin eri vaiheissa eivät ole ennustettavissa. Väitteet ovat siis osittain ristiriidassa keskenään. Jotta muutoksiin voidaan varautua, mallin toteutusvaiheessa tarvitaan tietoa siitä, minkä tyyppisiä mahdolliset muutokset ovat ja millä todennäköisyydellä niitä esiintyy. Mikäli tutkimustiedolla voidaan yhdistää erilaisia kohdetyyppisiä ja muutostarpeita, voidaan parametrin mallin suunnitteluvaiheessa tehdä päätöksiä, joilla näihin muutostarpeisiin varaudutaan mahdollistamalla muutosten toteuttaminen parametrisella mallilla.

Tässä työssä esiteltiin myös yhteistyömahdollisuuksia muiden suunnittelualojen kanssa hyödyntämällä parametrin mallia. Myös tällä saralla tutkimustiedossa on selkeä puute. Yhteistyössä on tunnistettavissa monia haasteita muun muassa tiedonsiirron ja rajapintojen osalta. Useiden suunnittelijoiden välisestä yhteistyöstä tarvitaankin kokemuseräistä tietoa esimerkiksi case-tutkimuksien avulla. Kokemuseräisen tiedon avulla päästään selvälle parametrin suunnitteluprosessin yhteistyön haasteista, ja edelleen kehittämään ratkaisua näille.

Optimoinnin osalta työssä esitettiin kokonaan uusi tapa, jolle ei löytynyt suoraa vastinetta kirjallisuudesta. Vaikka käytetyt optimointimenetelmät ja -algoritmit esitellään kirjallisuudessa, poikkeaa optimointimuuttujien valinta kirjallisuudesta esitetyistä menetelmistä. Tarkasteltu menetelmä on käytännönläheisempi, ja se imitoi suunnittelijan toimintaa alustavaa suunnitelmaa paranneltaessa. Tutkimuksen perusteella optimointimenetelmä antoi lupaavia tuloksia erityisesti optimointiin kuluvan ajan ja tuloksen käyttökelpoisuuden suhteen. Tutkimuksessa tarkastellut otoskoot olivat kuitenkin suhteellisen pieniä, eikä niitä näin ollen voi suoraan käyttää tulosten yleistämiseen. Lisäksi tutkimuksessa tarkasteltiin optimointia ainoastaan kahdella erilaisella numeerisella menetelmällä. Optimointimenetelmien soveltuvuutta esitetyn kaltaisiin tehtäviin on syytä tarkastella laajemmin. Mikäli esitetty optimointimenetelmä osoittautuu käyttökelpoiseksi, tulee tarkastella erityisesti parametrien valintaperusteita. Millaiset parametrit ovat optimoinnin kannalta tehokkaita? Millaisia parametreja on järkevää lukita, ja mitä arvoja niille tulee antaa? Näihin kysymyksiin vastaamalla voidaan parametrin suunnittelun yhteydessä toteutettavan optimoinnin tehokkuutta parantaa merkittävästi.

LÄHTEET

Aberdeen Group. (2007). The Design Reuse Benchmark Report: Seizing the Opportunity to Shorten Product Development. Massachusetts, Aberdeen Group.

Akos, G., Parsons, R., Payne, A. (2015). Foundations: The Grasshopper Primer Third Edition. Third Edition V3.3. ModeLab.

Archdaily. Queen Elizabeth Quay Bridge / Arup Associates. Saatavissa (viitattu 10.7.2017): <http://www.archdaily.com/786420/queen-elizabeth-quay-bridge-arup-associates>

Arora, J. (2004). Introduction to optimum design. Academic Press.

Autodesk. Dynamo. Saatavissa (viitattu 20.11.2017): <https://www.autodesk.com/products/dynamo-studio/overview>.

Avern-Taplin, A., Birmingham, N., Buxton, S. & Riddle, C. (2016). Elizabeth Quay Pedestrian and Cyclist Bridge, Australia. The Arup Journal, Issue 1/2016.

Bouma, W., Fudos, I., Hoffmann, C., Cai, J. & Paige, R. (1995). Geometric constraint solver, Computer-Aided Design, Vol. 27(6), pp. 487-501.

CURT. (2004). Collaboration, integrated information and the project lifecycle in building design, construction and operation. Construction Users Roundtable (CURT), Cincinnati, OH.

Davis, D. (2013). Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture, RMIT University.

Gamma, E. (1995). Design patterns: elements of reusable object-oriented software. Pearson Education India.

Gantt, M., & Nardi, B. A. (1992, June). Gardeners and gurus: patterns of cooperation among CAD users. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 107-117). ACM.

Glover, F. (1989). Tabu search—part I. ORSA Journal on computing, 1(3), 190-206.

Haapio, J. (2012). Feature-based costing method for skeletal steel structures based on the process approach. Tampereen teknillinen yliopisto. Julkaisu-Tampere University of Technology. Publication; 1027.

Hoffmann, C.M. & Joan-Arinyo, R. (2005). A brief on constraint solving, Computer-Aided Design and Applications, Vol. 2(5), pp. 655-663.

Horowitz, E. & Sahni, S. (1978). Fundamentals of computer algorithms, Computer Science Press.

Jalkanen, J. (2007). Tubular truss optimization using heuristic algorithms, Tampere University of Technology.

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1987). Optimization by simulated annealing. In *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications* (pp. 339-348).

Lamberti, L. (2008). An efficient simulated annealing algorithm for design optimization of truss structures, *Computers and Structures*, vol. 86, no. 19, pp. 1936-1953.

Lenz, M., Schmid, H. A., & Wolf, P. F. (1987). Software reuse through building blocks. *IEEE Software*, 4(4), 34-42.

Lim, W. C. (1994). Effects of reuse on quality, productivity, and economics. *IEEE software*, 11(5), 23-30.

McNeel North America. (2017). The Morpheus Hotel: From Design to Production: Live Webinar, webinaari. Saatavissa (viitattu 19.10.2017): <https://vimeo.com/203509846>.

Mela, K. (2013). Mixed Variable Formulations for Truss Topology Optimization. (Tampere University of Technology. Publication; No. 1134). Tampere: Tampere University of Technology.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6), 1087-1092.

Myung, S. & Han, S. (2001). Knowledge-based parametric design of mechanical products based on configuration design method, *Expert Systems with Applications*, Vol. 21(2), pp. 99-107.

Pavlovčič, L., Krajnc, A., & Beg, D. (2004). Cost function analysis in the structural optimization of steel frames. *Structural and Multidisciplinary Optimization*, 28(4), 286-295.

Pfafferott, J., Herkel, S. & Jäschke, M. (2003). Design of passive cooling by night ventilation: evaluation of a parametric model and building simulation with measurements, *Energy and Buildings*, Vol. 35(11), pp. 1129-1143.

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm intelligence*, 1(1), 33-57.

Preisinger, C. (2016). Karamba – parametric structural modeling, User manual for Version 1.2.2..

RIL 205-1-2017. (2017). Puurakenteiden suunnitteluohje - eurokoodi EN 1995-1-1. Suomen rakennusinsinöörien liitto RIL ry.

- Rutten, D. (2010). Evolutionary Principles applied to Problem Solving. Saatavissa (viitattu 15.11.2017): <http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles>.
- Rutten, D. (2012). Programming, conflicting perspectives. Saatavissa (viitattu 12.7.2017): <https://ieatbugsforbreakfast.wordpress.com/2012/04/01/programming-conflicting-perspectives/>.
- SFS-EN 1990. (2002). Eurocode. Rakenteiden suunnitteluperusteet. Suomen standardoimisliitto.
- SFS-EN 1992-1-1. (2004). Eurokoodi 2. Betonirakenteiden suunnittelu. Osa 1-1. Yleiset säännöt ja rakennuksia koskevat säännöt. Suomen standardoimisliitto.
- SFS-EN 1993-1-1. (2005). Eurocode 3. Teräsrakenteiden suunnittelu. Osa 1-1. Yleiset säännöt ja rakennuksia koskevat säännöt. Suomen standardoimisliitto.
- SFS-EN 1993-1-8. (2005). Eurocode 3. Teräsrakenteiden suunnittelu. Osa 1-8. Liitosten mitoitus. Suomen standardoimisliitto.
- SFS-EN 1991-1-5 + AC. (2006). Eurokoodi 5. Puurakenteiden suunnittelu. Osa 1-1. Yleiset säännöt ja rakennuksia koskevat säännöt. Suomen standardoimisliitto.
- Smith, R. (2007). Technical Notes from experiences and studies in using Parametric and BIM architectural software.
- Summers, J., & Buxton, S. (2017). Elizabeth Quay pedestrian bridge, Perth: timeless and refined in its simplicity. In Austroads Bridge Conference, 10th, 2017, Melbourne, Victoria, Australia.
- Sutherland, I.E. (1964). Sketchpad a man-machine graphical communication system, Transactions of the Society for Computer Simulation, Vol. 2(5), pp. R-3-R-20.
- Tanska, T. & Österlund, T. (2014). Algoritmit puurakenteissa: menetelmät, mahdollisuudet ja tuotanto, B 32, 1st ed. DigiWoodLab, Oulun yliopisto, Arkkitehtuurin tiedekunta.
- Tedeschi, A. (2014). AAD Algorithms-aided design: Parametric strategies using Grasshopper, Edizioni Le Pensur.
- Whitley, D. (1994). A genetic algorithm tutorial. Statistics and computing, 4(2), 65-85.
- Woodbury, R. (2010). Elements of parametric design, Taylor and Francis.
- Zaha Hadid Architects. Saatavissa (viitattu 8.11.2017): <http://www.zaha-hadid.com/interior-design/city-of-dreams-hotel-tower-cotai-macau/>.

LIITE A: OPTIMOINTIAJOJEN TULOKSET

Geneettinen algoritmi																
Setting	Parameters						Results					Average Results				
	Max stagnant	Division factor	BC division factor	BC bias	Number of divisions	Verticals	Comps	Mass [kg]	Iterations	TC_util	BC_util	max_util	Mass [kg]	Mass STD	Iterations	Iter. STD
Start		-2.50	0.00	0.00	3	0	1	2574.7		0.945	0.954	0.997				
5	-0.27	0.13	-0.25		3	1	1	1706.3	24	0.954	0.942	0.999				
5	-0.30	0.29	-0.70		4	1	1	1738.7	18	0.984	0.955	0.984				
5	0.43	0.14	-0.29		4	1	1	1710.8	19	0.980	0.958	0.980				
5	0.45	0.15	-0.29		5	0	2	1777.2	8	0.994	0.967	0.996				
5	0.95	0.19	-0.74		4	1	1	1710.0	33	0.968	0.958	0.968	1728.6	30.10	20.4	9.13
7	0.07	0.08	-0.52		3	1	1	1647.8	33	0.991	0.944	0.995				
7	-0.51	0.16	-0.27		3	1	1	1705.4	16	0.987	0.939	0.999				
7	0.40	0.05	-0.81		3	1	1	1702.1	36	0.947	0.946	0.996				
7	0.33	-0.07	-0.82		5	1	2	1742.5	13	0.975	0.967	0.999				
7	0.84	0.18	-0.53		4	1	1	1710.2	24	0.973	0.958	0.973	1701.6	34.12	24.4	10.11
10	0.35	0.25	-0.97		4	1	1	1709.1	92	0.999	0.957	0.999				
10	0.45	0.15	-0.46		4	1	2	1736.5	15	0.982	0.957	0.996				
10	0.55	0.15	-0.77		4	1	1	1710.3	34	0.971	0.958	0.970				
10	0.34	0.13	-0.59		4	1	1	1710.7	33	0.977	0.958	0.977				
10	-0.56	0.17	-0.93		3	1	1	1704.3	95	0.937	0.938	0.999	1714.2	12.74	53.8	37.04
15	0.34	0.22	-0.77		4	1	1	1710.5	30	0.999	0.957	0.999				
15	0.00	0.10	-0.71		3	1	1	1630.9	32	0.995	0.944	0.999				
15	0.17	0.20	-0.73		4	1	1	1710.9	51	0.998	0.957	1.000				
15	0.43	0.05	-0.49		3	1	1	1702.1	39	0.936	0.947	0.997				
15	0.20	0.12	-0.57		4	1	1	1710.8	45	0.987	0.957	0.987	1693.0	34.94	39.4	8.79
25	1.13	0.21	-0.67		4	1	1	1710.0	55	0.968	0.959	0.970				
25	0.93	0.19	-1.00		4	1	1	1709.7	69	0.984	0.958	0.984				
25	0.44	0.05	-0.92		3	1	1	1702.0	75	0.947	0.947	0.997				
25	0.47	0.04	-0.24		3	1	1	1702.8	109	0.934	0.947	0.996				
25	0.93	0.19	-0.99		4	1	1	1709.7	131	0.958	0.958	0.960	1706.8	4.06	87.8	31.26

Geneettinen algoritmi (BC bias = 0)

Setting	Max stagnant	Division factor	Parameters					Results			Average Results							Max stagnant			
			BC division factor	BC bias	Number of divisions	Verticals	Comps	Mass [kg]	Iterations	TC_util	BC_util	max_util	Mass [kg]	Mass STD	Iterations	Iter. STD	TC_util		BC_util	max_util	
Start		-2.50	0.00	0.00		3	0	1	2574.7		0.945	0.954	0.997								
	5	0.42	0.04	0.00		3	1	1	1702.9	32	0.932	0.947	0.994								
	5	0.38	0.15	0.00		5	0	2	1777.0	8	0.976	0.967	0.994								
	5	0.20	0.08	0.00		3	1	1	1647.9	17	0.994	0.946	0.999								
	5	0.44	0.07	0.00		4	1	2	1732.6	21	0.973	0.957	0.973								
	5	0.64	0.14	0.00		4	1	1	1711.6	5	0.980	0.958	0.980	1714.4	46.94	16.6	10.78	0.971	0.955	0.988	5
	7	0.22	0.07	0.00		3	1	1	1648.6	32	0.989	0.946	0.997								
	7	0.72	-0.20	0.00		5	1	2	1780.4	17	0.984	0.969	0.984								
	7	0.36	0.12	0.00		4	1	1	1711.6	13	0.980	0.958	0.980								
	7	0.44	0.14	0.00		4	1	1	1711.1	14	0.987	0.958	0.987								
	7	0.17	0.08	0.00		3	1	1	1647.9	20	0.996	0.945	0.998	1699.9	54.95	19.2	7.66	0.987	0.955	0.989	7
	10	0.87	0.18	0.00		4	1	1	1711.0	35	0.999	0.958	0.999								
	10	0.67	0.16	0.00		4	1	1	1711.0	36	0.993	0.958	0.993								
	10	0.42	0.04	0.00		3	1	1	1702.9	65	0.932	0.947	0.994								
	10	-0.44	0.15	0.00		3	1	1	1705.9	63	0.997	0.941	0.999								
	10	0.66	0.16	0.00		4	1	1	1711.0	21	0.993	0.958	0.993	1708.4	3.77	44.0	19.21	0.983	0.952	0.996	10
	15	0.42	0.04	0.00		3	1	1	1702.9	56	0.932	0.947	0.994								
	15	-0.44	0.15	0.00		3	1	1	1705.9	28	0.997	0.941	0.999								
	15	0.55	0.03	0.00		3	1	1	1703.5	42	0.932	0.948	0.995								
	15	0.32	0.13	0.00		4	1	1	1711.1	36	0.987	0.958	0.987								
	15	0.34	0.06	0.00		3	1	2	1687.9	47	0.999	0.948	0.999	1702.3	8.65	41.8	10.64	0.969	0.948	0.995	15
	25	0.17	0.08	0.00		3	1	1	1647.9	76	0.996	0.945	0.998								
	25	0.55	0.15	0.00		4	1	1	1711.0	61	0.990	0.958	0.990								
	25	0.17	0.08	0.00		3	1	1	1647.9	55	0.996	0.945	0.998								
	25	0.20	0.08	0.00		3	1	1	1647.9	85	0.994	0.946	0.999								
	25	-0.43	0.15	0.00		3	1	1	1705.9	52	0.997	0.941	0.999	1672.1	33.21	65.8	14.17	0.995	0.947	0.997	25

Geneettinen algoritmi (BC bias = 0, BC division factor = 0)

Setting	Division factor	Parameters					Results				Average Results							Max stagnant	
		BC division factor	BC bias	Number of divisions	Verticals	Comps	Mass [kg]	Iterations	TC_util	BC_util	max_util	Mass [kg]	Mass STD	Iterations	Iter. STD	TC_util	BC_util		max_util
Max stagnant	-2.5		0	0			2574.7		0.945	0.954	0.997								
Start					3	0													
5	0.76	0.00	0.00		3	1	1705.6	18	0.942	0.949	0.994								
5	0.87	0.00	0.00		3	1	1705.5	10	0.935	0.950	0.998								
5	0.83	0.00	0.00		3	1	1705.5	15	0.938	0.950	0.996								
5	0.83	0.00	0.00		3	1	1705.5	13	0.938	0.950	0.996								
5	0.83	0.00	0.00		3	1	1705.5	10	0.938	0.950	0.996	1705.52	0.044721	13.2	3.42053	0.9382	0.9498	0.996	
7	0.84	0.00	0.00		3	1	1705.5	14	0.937	0.950	0.997								
7	0.85	0.00	0.00		3	1	1705.5	12	0.937	0.950	0.997								
7	0.84	0.00	0.00		3	1	1705.5	13	0.937	0.950	0.997								
7	0.82	0.00	0.00		3	1	1705.5	12	0.939	0.950	0.996								
7	0.97	0.00	0.00		3	1	1705.6	10	0.947	0.949	0.991	1705.52	0.044721	12.2	1.48324	0.9394	0.9498	0.9956	
10	0.84	0.00	0.00		3	1	1705.5	17	0.937	0.950	0.997								
10	0.83	0.00	0.00		3	1	1705.5	16	0.938	0.950	0.996								
10	0.75	0.00	0.00		3	1	1705.6	15	0.914	0.949	0.994								
10	0.68	0.00	0.00		3	1	1705.6	20	0.918	0.949	0.991								
10	0.86	0.00	0.00		3	1	1705.5	16	0.936	0.950	0.998	1705.54	0.054772	16.8	1.92354	0.9286	0.9496	0.9952	
15	x	x	x	x	x	x	x	70	x	x	x								
15	x	x	x	x	x	x	x	70	x	x	x								
15	x	x	x	x	x	x	x	70	x	x	x								
15	x	x	x	x	x	x	x	70	x	x	x								
15	x	x	x	x	x	x	x	70	x	x	x	#DIV/0!	#DIV/0!	70	0	#####	#DIV/0!	#DIV/0!	15
25	x	x	x	x	x	x	x	115	x	x	x								
25	x	x	x	x	x	x	x	115	x	x	x								
25	x	x	x	x	x	x	x	115	x	x	x								
25	x	x	x	x	x	x	x	115	x	x	x								
25	x	x	x	x	x	x	x	115	x	x	x			115					25

Simuloitu jäähdytys

Setting	Number of Runs	Division factor	BC division factor	Parameters				Results				Average Results				Number of Runs		
				BC bias	Number of divisions	Verticals	Comps	Mass [kg]	TC_util	BC_util	max_util	Mass [kg]	Mass STD	Mass minimum	TC_util		BC_util	max_util
Start		-2.50		0.00	0.00	3	0	1	2574.7	0.945	0.954	0.997						
5	5	-1.59		0.11	-0.87	3	1	1	1733.3	0.998	0.927	0.998						
5	5	-0.01		0.10	-0.20	3	1	1	1707.4	0.921	0.944	1.000						
5	5	-1.31		0.12	-0.55	3	1	1	1732	0.991	0.930	0.991						
5	5	-0.97		-0.07	-0.01	4	1	2	1732.3	1.019	0.951	1.019	1723.9	12.08	1707.4	0.978	0.942	0.997
5	5	1.70		0.16	-0.06	4	1	1	1714.6	0.962	0.959	0.979						
7	7	-1.21		0.03	-0.12	3	1	1	1729.8	0.999	0.931	0.999						
7	7	-1.63		0.32	-0.98	3	1	1	1792.6	0.972	0.926	0.999						
7	7	0.05		0.06	-0.05	3	1	1	1649.3	0.993	0.944	0.993						
7	7	0.44		0.14	-0.13	4	1	1	1710.9	0.981	0.958	0.981	1724.3	51.74	1649.3	0.988	0.943	0.993
7	7	-0.38		0.28	-0.97	4	1	1	1738.7	0.994	0.955	0.994						
10	10	0.19		0.23	-0.58	4	1	1	1763.8	0.951	0.957	0.999						
10	10	1.35		0.03	-0.11	4	1	1	1733	0.964	0.960	0.964						
10	10	-1.34		0.15	-0.78	3	1	1	1730.4	0.992	0.930	0.992						
10	10	0.46		0.03	-0.59	3	1	1	1703.5	0.946	0.947	0.992						
10	10	-1.38		0.32	-0.15	4	1	1	1814.3	0.937	0.950	0.996	1749.0	42.29	1703.5	0.958	0.949	0.989
15	15	-0.63		0.13	-0.03	3	1	1	1707.5	0.990	0.938	0.990						
15	15	-0.26		0.13	-0.24	3	1	1	1706.3	0.954	0.942	1.000						
15	15	0.35		0.01	-0.04	3	1	1	1705.1	0.959	0.946	0.983						
15	15	-0.30		0.13	-0.07	3	1	1	1706.5	0.968	0.942	0.998						
15	15	-0.09		0.11	-0.51	3	1	1	1706.8	0.914	0.943	1.000	1706.4	0.88	1705.1	0.957	0.942	0.994
25	25	-0.26		0.13	-0.18	3	1	1	1706.3	0.958	0.942	1.000						
25	25	0.01		0.07	-0.16	3	1	1	1648.6	0.996	0.944	0.996						
25	25	0.86		-0.07	-0.19	3	1	1	1714.1	0.993	0.950	0.993						
25	25	-0.73		0.14	-0.18	3	1	1	1707.2	0.993	0.937	0.993						
25	25	-1.01		0.23	-0.98	3	1	1	1706.4	0.980	0.933	0.998	1696.5	26.98	1648.6	0.984	0.941	0.996

Simuloitu jäähdytys, (BC bias = 0)

Setting	Number of Runs	Division factor	BC division factor	Parameters			Results				Average Results					Number of Runs			
				BC bias	Number of divisions	Verticals	Comps	Mass [kg]	TC_util	BC_util	max_util	Mass [kg]	Mass STD	Mass minimum	TC_util		BC_util	max_util	
Start		-2.5		0		3		0	1	2574.7	0.945	0.954	0.997						
5	5	-0.56		0.08	0.00	3	1	1	1	1710.1	0.946	0.939	0.996						
5	5	-0.36		0.14	0.00	3	1	1	1	1706.2	0.985	0.941	0.999						
5	5	-0.92		0.54	0.00	4	1	1	1	1877.8	0.954	0.954	0.969						
5	5	-1.92		0.16	0.00	4	1	1	3	1782.4	0.998	0.946	0.998						
5	5	-1.01		0.10	0.00	3	1	1	1	1710.4	0.989	0.934	0.989	1757.4	74.48	1706.2	0.974	0.943	0.990
7	7	-1.63		0.03	0.00	4	1	1	2	1791.1	0.939	0.946	0.979						
7	7	-0.18		0.05	0.00	5	1	1	2	1758.8	0.979	0.966	0.989						
7	7	0.78		0.17	0.00	4	1	1	1	1711.0	0.996	0.958	0.996						
7	7	-0.09		0.01	0.00	5	1	1	2	1753.9	0.985	0.966	0.985						
7	7	0.77		0.01	0.00	3	1	1	1	1704.8	0.934	0.949	0.997	1743.9	35.92	1704.8	0.967	0.957	0.989
10	10	0.63		0.02	0.00	3	1	1	1	1704.2	0.935	0.948	0.995						
10	10	-0.61		-0.12	0.00	4	1	1	2	1781.3	0.998	0.953	0.998						
10	10	0.88		0.18	0.00	4	1	1	1	1711.0	0.998	0.958	0.998						
10	10	-0.28		0.13	0.00	3	1	1	1	1706.5	0.972	0.942	0.999						
10	10	-1.56		0.07	0.00	3	1	1	2	1756.5	0.995	0.928	0.995	1731.9	34.98	1704.2	0.980	0.946	0.997
15	15	0.04		0.05	0.00	3	1	1	1	1650.0	0.995	0.944	0.995						
15	15	-0.94		0.11	0.00	3	1	1	1	1709.6	0.993	0.935	0.993						
15	15	-1.52		0.05	0.00	3	1	1	1	1736.6	0.992	0.928	0.993						
15	15	0.98		-0.02	0.00	3	1	1	1	1710.0	0.955	0.950	0.995						
15	15	-0.09		0.10	0.00	3	1	1	1	1707.7	0.936	0.943	0.997	1702.8	31.83	1650.0	0.974	0.940	0.995
25	25	0.52		0.14	0.00	4	1	1	1	1711.3	0.984	0.958	0.984						
25	25	0.22		0.07	0.00	3	1	1	1	1648.6	0.989	0.946	0.997						
25	25	-0.36		0.14	0.00	3	1	1	1	1706.2	0.985	0.941	0.999						
25	25	1.16		0.20	0.00	4	1	1	1	1711.4	1.000	0.959	1.000	1697.0	27.16	1648.6	0.978	0.950	0.996
25	25	-0.04		0.10	0.00	3	1	1	1	1707.6	0.934	0.944	0.999						

Simuloitu jäähdytys, (BC bias = 0, BC division factor = 0)

Setting	Number of Runs	Division factor	BC division factor	Parameters				Results				Average Results					Number of Runs	
				BC bias	Number of divisions	Verticals	Comps	Mass [kg]	TC_util	BC_util	max_util	Mass [kg]	Mass STD	Mass minimum	TC_util	BC_util		max_util
Start		-2.5		0	0	3	0	1	2574.7	0.945	0.954	0.997						
5	5	-0.69		0.00	0.00	3	1	1	1730.9	0.987	0.937	0.987						
5	5	0.62		0.00	0.00	3	1	1	1705.7	0.950	0.948	0.989						
5	5	0.66		0.00	0.00	3	1	1	1705.6	0.948	0.949	0.991						
5	5	-0.19		0.00	0.00	3	1	1	1730.2	0.984	0.942	0.984						
5	5	0.11		0.00	0.00	3	1	1	1729.9	0.965	0.944	0.973	1720.5	13.52	1705.6	0.967	0.944	0.985
7	7	-0.51		0.00	0.00	3	1	1	1730.6	0.975	0.938	0.975						
7	7	0.85		0.00	0.00	3	1	1	1705.5	0.937	0.950	0.997						
7	7	-0.87		0.00	0.00	3	1	1	1731.2	0.999	0.935	0.999						
7	7	0.87		0.00	0.00	3	1	1	1705.5	0.935	0.950	0.998						
7	7	-0.20		0.00	0.00	3	1	1	1730.2	0.985	0.941	0.985	1720.6	13.79	1705.5	0.966	0.943	0.991
10	10	0.44		0.00	0.00	3	1	1	1705.8	0.961	0.947	0.983						
10	10	-0.01		0.00	0.00	3	1	1	1730.0	0.973	0.943	0.973						
10	10	-0.62		0.00	0.00	3	1	1	1730.8	0.982	0.937	0.982						
10	10	0.41		0.00	0.00	3	1	1	1705.8	0.963	0.947	0.982						
10	10	0.87		0.00	0.00	3	1	1	1705.5	0.935	0.950	0.998	1715.6	13.53	1705.5	0.963	0.945	0.984
15	15	0.83		0.00	0.00	3	1	1	1705.5	0.938	0.950	0.996						
15	15	0.86		0.00	0.00	3	1	1	1705.5	0.936	0.950	0.998						
15	15	0.66		0.00	0.00	3	1	1	1705.6	0.948	0.949	0.991						
15	15	0.56		0.00	0.00	3	1	1	1705.7	0.954	0.948	0.987						
15	15	0.85		0.00	0.00	3	1	1	1705.5	0.937	0.950	0.997	1705.6	0.09	1705.5	0.943	0.949	0.994
25	25	0.86		0.00	0.00	3	1	1	1705.5	0.936	0.950	0.998						
25	25	0.84		0.00	0.00	3	1	1	1705.5	0.937	0.950	0.997						
25	25	0.85		0.00	0.00	3	1	1	1705.5	0.937	0.950	0.997						
25	25	0.82		0.00	0.00	3	1	1	1705.5	0.939	0.950	0.996						
25	25	0.67		0.00	0.00	3	1	1	1705.6	0.947	0.949	0.991	1705.5	0.04	1705.5	0.939	0.950	0.996